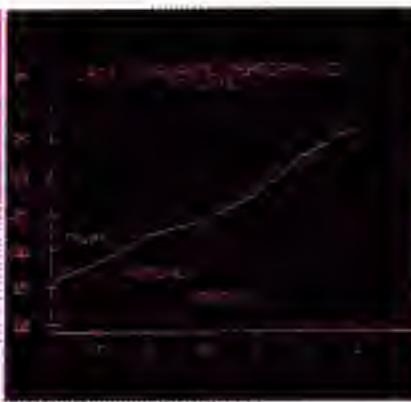




GRAPHICS TABLET™

OPERATION AND REFERENCE MANUAL



<http://www.turbo-2.com/apple/>

NOTICE

Apple Computer Inc. reserves the right to make improvements in the product described in this manual at any time and without notice.

DISCLAIMER OF ALL WARRANTIES AND LIABILITY

APPLE COMPUTER INC. MAKES NO WARRANTIES, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS MANUAL OR WITH RESPECT TO THE SOFTWARE DESCRIBED IN THIS MANUAL, ITS QUALITY, PERFORMANCE, MERCHANTABILITY, OR FITNESS FOR ANY PARTICULAR PURPOSE. APPLE COMPUTER INC. SOFTWARE IS SOLD OR LICENSED "AS IS". THE ENTIRE RISK AS TO ITS QUALITY AND PERFORMANCE IS WITH THE BUYER. SHOULD THE PROGRAMS PROVE DEFECTIVE FOLLOWING THEIR PURCHASE, THE BUYER (AND NOT APPLE COMPUTER INC., ITS DISTRIBUTOR, OR ITS RETAILER) ASSUMES THE ENTIRE COST OF ALL NECESSARY SERVICING, REPAIR, OR CORRECTION AND ANY INCIDENTAL OR CONSEQUENTIAL DAMAGES. IN NO EVENT WILL APPLE COMPUTER INC. BE LIABLE FOR DIRECT, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY DEFECT IN THE SOFTWARE, EVEN IF APPLE COMPUTER INC. HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. SOME STATES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF IMPLIED WARRANTIES OR LIABILITY FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THE ABOVE LIMITATION OR EXCLUSION MAY NOT APPLY TO YOU.

This manual is copyrighted. All rights are reserved. This document may not, in whole or part, be copied, photocopied, reproduced, translated or reduced to any electronic medium or machine readable form without prior consent, in writing, from Apple Computer Inc.

©1979 by APPLE COMPUTER INC.
10260 Bandley Drive
Cupertino, California 95014
(408) 996-1010

All rights reserved.
APPLE product # 030-0076-00

GRAPHICS TABLET™

OPERATION AND REFERENCE MANUAL

TABLE OF CONTENTS

INTRODUCTION

CHAPTER 1 GETTING STARTED

4	What You Will Need
5	Unpacking
6	Plugging In
7	Installing the Interface
9	Backing Up the Diskette
9	Starting Up
10	The menu Overlay
10	Aligning the menu

CHAPTER 2 THE GRAPHICS TABLET SOFTWARE

14	Get Ready
14	Drawing
15	The Menu
15	Delta
17	The Color Menu
18	Some background Information
18	A brief Digression on High-Resolution Graphics
19	Guidelines
19	A Clear Alternative
20	Line Up
21	Draw
21	Yes, Sir, Dot's My Baby
22	The Big Frame Up
23	Little Boxes
24	Saving Pictures for Posterity
27	Browsing Through the Catalog
28	Getting Loaded
29	Getting Out
30	Room With a Viewport
33	A Brilliant Reduction

34	Opening the Window
36	Broken Windows
36	Drawing in the Window
37	Reset
36	A Softer Reset
38	Calibrate
40	Mistaken Calibration
40	Long Distance...
41	...And Area Codes
42	Slide Rules
43	Prismatic Apple
44	In Conclusion

CHAPTER 3

PROGRAMMING THE GRAPHICS TABLET

46	The Programs
47	Tablet-Code Applesoft
49	The Main Loop
50	Menu Items
51	Extending the Menu
51	Example: Instant Color Menu
53	Example: Circle Mode
55	Example: Disc Mode
56	The Firmware
57	Tablet Control
60	Accepting Input
61	From Machine Language
64	Quick-Draw
65	By Any Other Name

APPENDIX A

USE AND CARE

68	Care of the Menu Overlay
68	Care of the Tablet
69	Care of the Interface
69	If It Doesn't Work

APPENDIX B

COPYING THE MASTER DISKETTE

72	With Two Disk Drives
72	With One Disk Drive

APPENDIX C

COLOR ANOMALIES

76	Unusual Color Effects...
77	...And How to Get Them

APPENDIX D

PROGRAM LISTINGS

80	Tablet-Code Applesoft
85	Variable Atlas
87	Subroutines
88	Special Locations
89	ROM Code
108	Quick-Draw
118	Utilities

APPENDIX E

SCHEMATIC DIAGRAM

123	
-----	--



INTRODUCTION

Welcome to the world of the Apple Graphics Tablet. The Tablet is a device which converts the position and movements of a special pen into numbers which your Apple can use and understand. The programs, or "software", supplied with your Tablet tell the Apple how to draw pictures on its high-resolution graphics screen, using the information supplied by the Tablet. These programs turn your Apple and Tablet into an artist's sketchpad, an engineer's drawing board, or a mathematician's chalkboard. With the Tablet and the supplied programs you can draw freehand pictures on the Apple's screen, or use the Apple to draw straight lines, rectangular boxes, open frames, or tiny dots. The pictures you create can easily be saved on Apple diskettes and recalled anytime you want. You can use the Apple to calculate the areas and distances of shapes and lines you draw on the Tablet, and you can change the scale of the figures you draw.

This is the Operation and Reference Manual for the Apple Graphics Tablet. The Graphics Tablet is a "hands-on" product, and the best way to learn how to use it is to take pen in hand and start experimenting. Most of this book is based on the assumption that you have the Graphics Tablet set up in front of you, and are following and doing each example as it is presented. If you try to learn how to use the Tablet without using this manual (or even worse, read the manual without actually using the Tablet), you might pick up most of the simpler commands, but you'll never master the more powerful functions of the Tablet. So read the manual, repeat the examples, and don't be afraid to experiment.

The first chapter of this book describes how to set up your Tablet, and what you need in order to use it. Chapter 2 introduces you to the Graphics Tablet software. This is a set of programs which allow you to use the Tablet to draw pictures on the Apple's high-resolution graphics screen. You do not need to know much about the Apple in order to use the Graphics Tablet. In fact, all you really need to know is how to turn it on. Once you start using the Tablet software, it will guide you each step of the way. You do not need to know how to write programs to use the Tablet skillfully and efficiently.

If you do know how to program, you may be interested in Chapter 3. It will give you assistance in modifying the Graphics Tablet software to your liking, including adding your own features to the Tablet menu. There are also instructions on interfacing directly to the Tablet's firmware, so you can write your own special-purpose programs that will use the Tablet. Listings of the programs which operate the Tablet are supplied in Appendix D.



If you see the symbol



it means that the following paragraph contains important information about some Tablet behavior that you might not anticipate. The symbol



means that the following paragraph contains special information you should note. Read these sections carefully.

Above all, feel free to play around with the Tablet. The Apple Graphics Tablet is easy to learn, easy to use, and hard to mess up. With some simple maintenance (described in Appendix A), your Tablet will give you years of enjoyment and use. So sit down at your Apple, take pen in hand, and turn to Chapter 1. We'll let you...

Draw Your
Own
Conclusion!

CHAPTER 1 GETTING STARTED

4	What You Will Need
5	Unpacking
6	Plugging In
7	Installing the Interface
9	Backing Up the Diskette
9	Starting Up
10	The Menu Overlay
10	Aligning the Menu

WHAT YOU WILL NEED

To use the Apple Graphics Tablet with its supplied software, you will need the following:

- 1) An Apple II or Apple II Plus computer, with 48K bytes of Random Access Memory (RAM);
- 2) If you do not have an Apple II Plus, you will need an Applesoft Firmware card (part number A2B0009), or an Apple Language System (part number A2B0006) with a BASICS language diskette;
- 3) An Apple Disk II plug-in controller card with at least one Disk II disk drive;
- 4) A color or black-and-white video monitor.

In addition, you may wish to have additional Disk II disk drives and controller cards.



The Graphics Tablet was designed to work with most present and future Apple II hardware and software. However, the supplied programs which operate the Graphics Tablet are designed to work with the Apple II DOS disk operating system, versions 3.2 and up. The Graphics Tablet software will not operate under previous versions of DOS or in an Apple Pascal environment.

It is helpful (but not necessary) to have read the following manuals:

- 1) The Applesoft Tutorial (product number A2L0018)
Welcome and Chapter 1
- 2) Do's and Don't's of DOS (product number A2L0012)
Preface through Chapter 2

If you are using the Apple Language System, be sure to read:

Apple Language System (product number A2L0024)
Chapter 3: Using BASIC

UNPACKING

Your Graphics Tablet package contains ten items:

- 1) The Graphics Tablet and its attached cable.
- 2) The Graphics Tablet's indicator pen and its attached cable.
- 3) A printed-circuit board (the Graphics Tablet Interface card)
- 4) A mylar "menu" overlay.
- 5) Two "GRAPHICS TABLET SOFTWARE" diskettes.
- 6) A piece of die-cut, double-sided foam tape.
- 7) A warranty card.
- 8) A packing list.
- 9) This manual.
- 10) A static cloth.

Save the packing material in case you wish to transport your Tablet -- or in the unlikely event that you must return your Tablet to your dealer for service. If you did not fill out your warranty with your Apple dealer before you brought your Graphics Tablet home, send it in now -- not only does this ensure that any warranty repair your Tablet may need will be done as quickly as possible, but it also puts you on the mailing list for CONTACT, the Apple users' newsletter that keeps you informed of updates and new products.

PLUGGING IN

***** Special Note *****
Before connecting or disconnecting
ANYTHING
on the Apple or
the Graphics Tablet
TURN OFF THE POWER.
This is a must.

Please pay special attention to this warning. If you try to connect or disconnect something from the inside of your Apple when the power is on, there is a good chance that you may damage its electronics.

The Graphics Tablet and its pen connect to the Interface card, which in turn plugs into one of the eight peripheral connector slots in the inside of the Apple, along the back of the main board. The cables attached to the Tablet and the pen terminate in small sockets, which fit over two sets of pins on the Interface card. The sockets are spaced and keyed so that it is very difficult to attach them incorrectly.



Handle the Interface card as you would handle a high-quality, expensive phonograph record. Grasp it only by the corners or edges, and try not to touch the delicate components or pins. Don't grasp the card by the gold "fingers" -- they are the medium through which the Apple communicates to the Tablet and their efficiency is decreased if they are dirty or scratched. The Interface is a precision instrument and should be treated with care.



First attach the Tablet's pen to the interface card. Place the Interface card on a flat surface with the components face up and the gold "fingers" nearest you. Take the connector at the end of the cable from the Tablet's pen. Notice that the four tiny round holes on the bottom of the connector are keyed to correspond to the set of four pins in the upper right corner of the Interface card. Gently slide this connector over the set of pins. There should be some space between the card and the connector. The finished connection should look like this:



Now attach the Graphics Tablet to the Interface card. Take the connector at the end of the cable from the Graphics Tablet. Gently slide the connector over the set of pins near the top middle of the Interface card. When the connector is properly attached there should be some space between it and the card. The finished connection should look like this:



INSTALLING THE INTERFACE



To install the Graphics Tablet Interface card (which you have already connected to the Tablet and its pen) into the Apple, you will simply plug the Interface card into the back of the computer, as follows:



1. Turn off the power switch at the back left corner of the Apple. This is important to prevent damage to the computer. Don't unplug the Apple, just turn it off. If you unplug your Apple, you isolate it from the common earth ground and your Apple and Tablet Interface card could be in danger from static discharges.



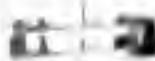
2. Remove the cover from the Apple. Do this by pulling up on the back edge of your Apple's lid until the corner fasteners pop apart, then slide the lid back and lift it off.



3. Before proceeding, touch your hand to the metal power supply case inside your Apple. This will remove any stray static charges from your hands, so you do not damage the static-sensitive components on the Interface card.



4. Inside the Apple, across the rear of the main green board, are eight long, narrow sockets called Peripheral Connectors, or "slots". The leftmost slot (looking from the keyboard end) is called "Slot #0" and the rightmost is called "Slot #7". The Interface card will operate in any slot except #0, but it is customary for the Tablet to use Slot #5, the third one from the right.



5. Grasp the upper corners of the card between the thumbs and forefingers of both hands. Insert the gold "fingers" of the Interface card into the chosen slot in the back of the Apple, rear edge first. Gently push the front edge of the card down until it is level and firmly seated.

6. Take the two cables which you have connected to the Interface card. On the cable attached to the pen there is a black plastic fitting. This is called a strain relief. There is a hole running lengthwise along the bottom of the strain relief, with a slit running the length of the hole. Pry the slit open with your fingernail and slide the cable from the Tablet through the slit and into the hole. The finished strain relief should look like this:



Now take the strain relief with its "tail" on top and pointing out the back of the Apple and slide it into the leftmost of the two smaller vertical notches in the back of the Apple's case. Slide it down to the bottom of the notch. It should be a tight fit. If it doesn't slide all the way down the first time, pull it out and slide it back in again. The plastic is pliable enough so that it will conform to the slot's width after about three or four insertions.



7. Snap the top back onto your Apple. Place the Tablet on a flat surface near your Apple, close enough so that the pen can easily reach all parts of the Tablet surface. Make sure that your disk drive and video monitor are connected properly.

BACKING UP THE DISKETTE

Now that your Graphics Tablet is all hooked up, it's a good time to think about an important rule of thumb. "What rule?" you might ask. The rule is this: Always keep at least one backup copy of any diskette whose information you wish to keep.

The value of a backup copy cannot be overemphasized. Right now, if you were to drop both your Graphics Tablet Software diskettes, and your pet turtle started nibbling on them, or somebody mistook them for square, black Frizbees, or some other catastrophic event occurred which would render them both unreadable, then your Graphics Tablet would be almost useless. Honest. You'd have to write all new programs yourself, or buy another Graphics Tablet Software diskette, in order to use your Tablet.

Take a look at the two Graphics Tablet Software diskettes that came with your Graphics Tablet. Notice that one of them has a small piece of silver tape over the rectangular notch on its edge. This piece of tape is called a write-protect tab. The write-protect tab tells the Apple not to store any more information on the diskette in question. The tab assures that none of the information on the diskette will be accidentally written over. Store this write-protected diskette in a safe place, and use it as your backup copy.

Fortunately, you know better than to leave your Graphics Tablet diskettes lying around where they might be damaged by heat, your pet turtle, or strong magnetic fields. However, you may want to be really careful and keep two backup copies instead of just one. Keeping more than one backup copy insures that your programs will be safe even if one of your backups is accidentally destroyed. If you don't know how to go about making copies of the Graphics Tablet Software diskette, see Appendix B in this manual for instructions.



Don't put your Graphics Tablet Software diskettes, or any other diskettes, on top of the Tablet itself! Its magnetic field will wipe out any information on the diskettes.

STARTING UP

After you've reassembled your Apple and its peripherals and everything is in order, place your Graphics Tablet Software diskette into Drive 1. Remember to use the one that does not have the silver write-protect tab over the rectangular notch on its edge. Now turn the power on and "boot" the diskette. (If you don't understand what this means, STOP! Don't kick your diskette, but read the section called **BOOTING DOS** in Chapter 2 of your DOS manual, or Chapter 3 in the

Language System Manual if you have an Apple Language System.) The disk drive will whirr and click for about 15 seconds, then the Graphics Tablet logo will be displayed:



To begin your encounter with the Tablet, press the **ESC** key. The screen will display the Graphics Tablet "HELI0 Menu", which is a list of things you can do with your Graphics Tablet Software diskette. You'll be using the MENU ALIGNMENT program first.

THE MENU OVERLAY

Included in your Graphics Tablet package is a mylar overlay called the "Graphics Tablet Menu". You will be placing this overlay in the center of the recessed area on the Tablet. The overlay divides the surface of the Tablet into different areas, and each area has a different meaning. Part of the overlay represents the Apple's high-resolution graphics screen, and another part lets you select which functions of the Tablet you want to use.

Once you attach this overlay to the Tablet, you need to tell the Apple the exact location of the overlay on the surface of the Tablet, and the Apple will help you make sure that you've put the overlay on correctly.

ALIGNING THE MENU

Before you use the Tablet, you must first place the overlay on the Tablet and align it. There is a program on your diskette which will assist you in aligning your menu overlay properly. From the Graphics Tablet HELI0 menu, press **M** to select the MENU ALIGNMENT program, and then press **RETURN**.

The alignment program tells you what slot your Interface card is plugged into and then creates an information file on your diskette. The name of this file is TAB.INFORMATION. All other programs which use the Tablet can read the vital information about your Tablet and menu from this file. After you run the MENU ALIGNMENT program once, you need not run it again, unless you remove your menu overlay from the Tablet or use your Tablet with a different pen.



The MENU ALIGNMENT program will guide you in attaching and aligning the menu overlay. All you need to do is read its instructions carefully, and do just as it requests. If the menu ever comes loose from the Tablet during the alignment process, press the **[ESC]** key to re-start the whole procedure.



You'll be using four small circles of thin double-sticky foam tape, included with your Tablet, to attach the menu to the Tablet surface. Stick a small circle of tape directly under the target circle in the upper-left corner of the menu overlay, and place the overlay in the center of the recessed area of the Tablet. Stick the overlay to the Tablet surface.



Use the Graphics Tablet's pen to point to the small circle in the upper-left hand corner of the command box labelled RESET. Hold the pen perpendicular to the surface, and carefully press straight down until the point retracts into the pen, making sure that the point of the pen does not slip out of the circle.



Now take the pen and point to the small circle at the lower-left corner of the overlay. Hold the pen straight and press down. The Apple will now determine whether the overlay is straight or crooked. If it is straight, the screen will display "ALIGNED" and you can proceed. If the overlay is crooked, the program will ask you to swing the bottom edge of the overlay a little to one side. Move the overlay just a little in the proper direction and try again. Continue until the screen displays "ALIGNED". Place circles of tape under the remaining three target circles and stick them firmly to the Tablet surface. You may want to press **ESC** to end the program and then re-RUN it to make sure you didn't accidentally move the overlay when you were taping down the corners.



Now follow the arrows displayed on the screen and press the pen down in each small circle in all four corners of the overlay. Be very careful! Make sure that you're holding the pen straight up-and-down, and that the point of the pen does not stray outside of the target. If you do it correctly, you will be rewarded with the message



CREATING TABLET INFORMATION FILE

If you get any other message, you probably slipped somewhere, or the overlay isn't centered on the Tablet surface. Try it again.

Once the overlay is aligned, the Apple will return you to the Graphics Tablet logo. Press **esc** to get to the program menu again. Now you can start using your Graphics Tablet.

(If you want to be really sure that your menu is properly aligned, you can run the MENU ALIGNMENT program again. Leave the menu taped down and just poke the proper points with the pen. If everything goes well, then your menu is well-aligned. If not, repeat the MENU ALIGNMENT procedure.)

CHAPTER 2

THE GRAPHICS TABLET SOFTWARE

14	Get Ready
14	Drawing
15	The Menu
15	Delta
17	The Color Menu
18	Some Background Information
18	A Brief Digression on High-Resolution Graphics
19	Guidelines
19	A Clear Alternative
20	Line Up
21	Draw
21	Yes, Sir, Dot's My Baby
22	The Big Frame Up
23	Little Boxes
24	Saving Pictures for Posterity
27	Browsing Through the Catalog
28	Getting Loaded
29	Getting Out
30	Room With a Viewport
33	A Brilliant Reduction
34	Opening the Window
36	Broken Windows
36	Drawing in the Window
37	Reset
38	A Softer Reset
38	Calibrate
40	Mistaken Calibration
40	Long Distance...
41	...And Area Codes
42	Slide Rules
43	Prismatic Apple
44	In Conclusion

GET READY

To start using the Graphics Tablet, go to the Graphics Tablet logo, either by re-booting the diskette, completing the MENU ALIGNMENT program, or typing

RUN HELLO

Press **ESC**. Now press **G** to select GRAPHICS TABLET SOFTWARE and press **RETURN**. The disk will whirr and chug for a while, and the Apple will present you with a blank screen. (If you get a message informing you that the Tablet information file does not exist, press **[RETURN]** and run the MENU ALIGNMENT program.) In about three seconds, your Tablet will be ready to use.

DRAWING

Touch the point of the Tablet's pen lightly to the surface of the Tablet. Move the pen around. You should see a small "crosshairs" cursor moving around the screen as you slide the pen around. The crosshairs are a locator, and the position and motions of the crosshairs on the screen correspond to the position and motions of the pen on the Tablet. Now press down on the pen so that the point retracts, and start drawing. As you draw on the Tablet, the path you trace will show up on the screen as a thin white line.



The top and sides of the working area of the overlay (the area with the fine mesh gridwork) correspond to the top and two sides of the Apple's screen. However, the working area on the overlay is slightly taller than the screen. To compensate for this difference in height, only the upper 2/3 of the overlay's working area is "mapped onto" the screen. The rest, about 2.5 inches (6.35 cm) at the bottom of the working area, is not usually active. (For information on how to use



the full working area, see the WINDOW command.) You might want to find the lower boundary of the working area and mark it with a felt tipped pen on the overlay.



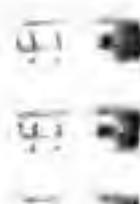
THE MENU

Along the top of the Tablet's Mylar overlay are two rows of 22 squares. Each square in the top row carries the name of a certain command or function which the Tablet software can perform. These two rows of squares are called the Tablet Menu. They let you order functions for the Tablet as you would order food in a restaurant in a foreign country: by pointing to what you want. If you could speak the proper language, you would order dinner by telling the waiter what you would like. But the Tablet's language consists of thousands of magnetic and electrical impulses traveling near the speed of light. Most people can't communicate in this fashion (those who can are mutants, and thus have gone far in the computer world), so you'll have to indicate your choices to the Tablet by pointing at the Menu.

To invoke a command or function, touch the point of the Tablet's pen anywhere inside the corresponding square and press down. Hold the pen down until you hear the Apple beep. If you don't hear a beep, then you haven't fully activated the command, and you should lift the pen and try again.

The second row of boxes, which carry no name, consequently have no function. You can use them for your own programs (see EXTENDING THE MENU in Chapter 3).

The following pages describe each command and its function. To help you locate the square for each command, the section describing that command will be headed with a drawing of the menu and a pen pointing to the proper square.



DELTA

RESET	CLEAR	WINDOW	BG COLOR	DELTA	SOFT RESET	VIEW PORT	CALIBRATE	REDUCER	PEN COLOR	DRAW
-------	-------	--------	----------	-------	------------	-----------	-----------	---------	-----------	------

The DELTA function lets you adjust the precision with which the pen draws on the screen. The Apple subdivides the working area of the Tablet into 53,760 small dots, each one corresponding to one dot on the Apple's screen. As you move the pen around the surface of the Tablet, the Apple draws lines between the dots you traverse. The DELTA setting lets you control the distance the pen can move before the Apple draws a line to the new dot. The smallest possible DELTA is 1. This setting will make the Apple draw a new line each time you move the pen a vertical or horizontal distance of one dot (or the

Tablet, about .039 inches or 0.997 mm) from the last dot plotted. The normal value for DELTA is 2. The largest DELTA value is 127. This will make the Tablet draw a new line only after the pen has moved a horizontal or vertical distance of 127 dots (4.98 inches, or 12.6 cm) from the last dot plotted.

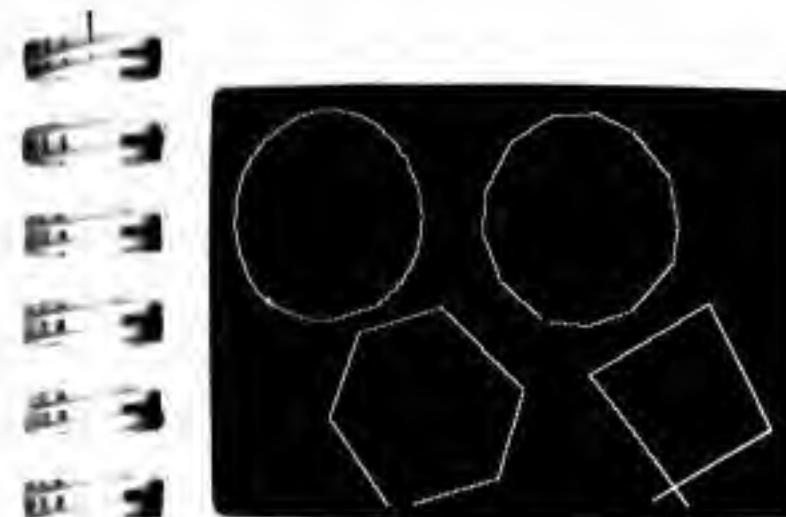
Associated with the DELTA setting is the Audio Feedback feature. When this feature is turned on, the Apple's speaker will emit a click each time the Apple draws a new line. With the Audio Feedback feature enabled, you can actually hear as well as see the effects of different DELTA settings.

To look at or change the current DELTA setting, touch the pen to the DELTA command square. Press it down until you hear the Apple beep. You'll see the following:



The first few lines tell you the current DELTA setting and whether the Audio Feedback feature is ON or OFF. The Apple will ask you for the new DELTA value. If you wish to retain the current DELTA value, just press **RETURN**. If not, type a number between 1 and 127 and press **RETURN**. Next, you'll be asked whether you want the Audio Feedback ON or OFF. Again, if you wish to retain the current setting, just press **RETURN**. Otherwise, type the word ON (to produce the clicks) or OFF (to silence the clicks) and press **RETURN**. The Apple will then return you to the picture you were drawing, with the new DELTA and Audio Feedback settings in effect.

Experiment a little with different DELTA settings. Set a cup or saucer on the Tablet surface and trace its perimeter several times, using different values for DELTA each time. You'll get something which looks like this:



Turn the Audio Feedback ON and OFF, and use it at different DELTA settings. At low settings, it will buzz as you move the pen around; at higher settings, you'll be able to detect distinct clicks.

THE COLOR MENU



The Apple's screen can display six colors: black, white, green, violet, orange, and blue (of course, if you are using a black-and-white monitor, you'll see only various shades of grey). The Apple lets you draw on the screen with all of these colors.

Touch the pen to the command box marked PEN COLOR and press down. The Apple will beep, the screen will clear and the message

CONSTRUCTING COLOR MENU

will appear at the bottom of the screen. The Apple will proceed to draw eight colored boxes, surrounded by a grey border.

Move the pen lightly across the surface of the Tablet. You'll see a small block drifting around the screen (instead of the usual crosshairs). Use the pen to position the block over the color with which you wish to draw, and press down. The color menu will vanish, and you will be looking at the screen on which you were previously drawing. Now, draw! The lines you draw will be in the color you selected. Change colors again and keep drawing. All the rules are the same. Only the colors have been changed.

If, while you're shopping around for a new PEN COLOR, you decide you really don't want to change the color you've got, just press **RETURN**. Your PEN COLOR will not be changed.

SOME BACKGROUND INFORMATION

RESET	CLEAR	WINDOW	BG COLOR	DELTA	SOFT RESET	VIEW PORT	CALI-BRATE	REDUCER	PEN COLOR	DRAW
-------	-------	--------	----------	-------	------------	-----------	------------	---------	-----------	------

When you start drawing with the Graphics Tablet, you're given a black screen on which to create. You can tell the Tablet that you wish to use a different-colored background by pressing the pen in the BG COLOR (BackGround COLOR) square. The Apple will present you with a color menu (as for the PEN COLOR command). Pick the color you want to use as a background; for instance, orange. The menu will vanish and the screen will instantly be filled with orange, or whatever color you have chosen.



Using the BG COLOR command will erase everything you had on the screen, so if you want to specify a BackGround COLOR, do it before you start to draw.

Are you trying the examples? Is the BackGround COLOR command working? Is orange your favorite color? Again, if you decide not to change the BackGround COLOR, just press **RETURN** instead of selecting a color. Your BackGround COLOR (and your picture, too) will be left unchanged.

A BRIEF DIGRESSION ON HIGH-RESOLUTION GRAPHICS

by now you must have noticed that there are some funny things going on with the colors. For example, set the BackGround COLOR to green and try to DRAW blue lines across it. Or set the BackGround COLOR to violet, and draw some blue lines. Obviously there's something wrong. The color "shadows" and the "zebra stripes" which you see on a color television set, or the strange distortions, unevenness, and lack of consistency you observe on a black-and-white monitor, are the results of the Apple's method of generating colors in its high-resolution graphics display. For more information on the anomalies of the Apple's high-resolution graphics color generation scheme, see Appendix C.

GUIDELINES

To minimize the problems created by the Apple's high-resolution graphics color scheme, follow these guidelines:

- 1) Most inconsistencies of the Graphics Tablet colors occur with vertical lines. Use horizontal lines when possible.
- 2) When you're drawing with black or white on a colored background, or in color on a black or white background, draw the lines a little thicker than normal by going over them twice. This takes care of the broken lines you may get.
- 3) If you need to place two colored blocks next to each other, stack them vertically, not horizontally. This cures the colored shadows that sometimes appear between colors.

So much for the digression, on with the Tablet.

A CLEAR ALTERNATIVE

RESET	CLEAR	WINDOW	BG COLOR	DELTA	SOFT RESET	VIEW PORT	CALI-BRATE	REDUCER	PEN COLOR	DRAW
-------	-------	--------	----------	-------	------------	-----------	------------	---------	-----------	------

If you're tired of the scribbles and doodles on your screen, press the pen to the CLEAR square. Zap! Your whole screen will be restored to the BackGround COLOR (see the previous section). Draw mode will be restored, and, if you haven't set one, the BackGround COLOR will be black.

If you have set a VIEWPORT (described a little further on in this chapter), then CLEAR will affect only the portion of the screen inside the VIEWPORT. The rest of the screen will remain unchanged.

LINE UP

LINES	DOTS	FRAME	BOX	CATALOG	LOAD	SAVE	SEPARATE	SLIDE	AREA	DISTANCE
-------	------	-------	-----	---------	------	------	----------	-------	------	----------

So far, you've been happily drawing somewhat rough, freehand lines on the Apple's screen. If you wanted to draw a straight line between two points, you probably tried to draw it with a straightedge (smart, but awkward) or did it freehand (sloppy). "Is there a better way to draw straight lines?", I hear you cry. Well, guess what! Yes, there's a better way to draw straight lines. Press the pen to the box which, for some obscure reason, bears the designation LINES. Now you have entered LINES mode. You will remain in LINES mode until you tell the Apple otherwise. We'll tell you how to do that later.

Meanwhile, since you're in LINES mode, let's draw some lines. Press the pen down anywhere on the Tablet's working area and lift the pen again. See the small dot left on the screen? That will be one endpoint of your line. Now press the pen down at another point in the working area. Zap! There's now a straight line connecting the two points. Press the pen down again at another point, and the Apple will draw another line, this one connecting the new point and the second point. Now rush to your nearest toy store and buy a Connect-the-Dots coloring book. Pick out an interesting page, tape it to your Tablet, and start connecting dots. The figure will magically appear on your screen.

If you want to start a second LINES figure, simply press the pen to the LINES command box again. The next point at which you press the pen will be the beginning of a new figure.



The "straight" lines you draw with your Tablet may not seem absolutely straight to you. This is normal. Lines that are neither horizontal nor vertical are actually made up of tiny zig-zags between dots on the screen.

Once you enter LINES mode, you'll stay in LINES mode until you ask to leave. The proper way to ask to be excused is to press the pen to a box that represents another drawing mode.

DRAW

RESET	CLEAR	WINDOW	BG COLOR	DELTA	SOFT RESET	VIEW PORT	CALIBRATE	REDUCER	PEN COLOR	DRAW
-------	-------	--------	----------	-------	------------	-----------	-----------	---------	-----------	------

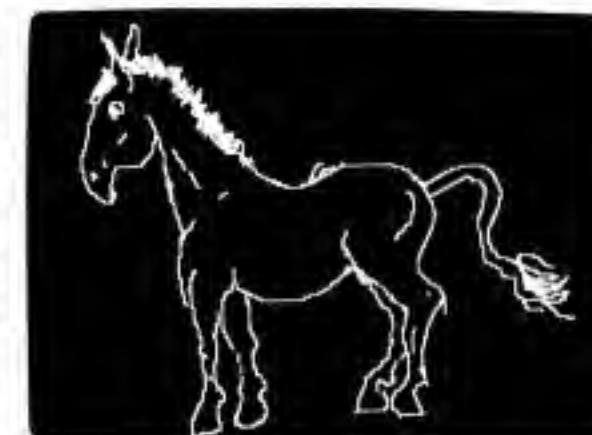
The mode you were in before you entered LINES mode is called DRAW mode. DRAW mode is the normal state of the Graphics Tablet and is automatically put into effect when you choose the Graphics Tablet Software from the diskette menu. This means that DRAW mode is the default mode.

Whenever you wish to leave a fancy drawing mode (LINES, BOX, FRAME or DOTS), simply press the pen to the command square called DRAW. Your picture will be left intact and you will be able to draw normally until you specify another mode.

YES, SIR, DOT'S MY BABY

LINES	DOTS	FRAME	BOX	CATALOG	LOAD	SAVE	SEPARATE	SLIDE	AREA	DISTANCE
-------	------	-------	-----	---------	------	------	----------	-------	------	----------

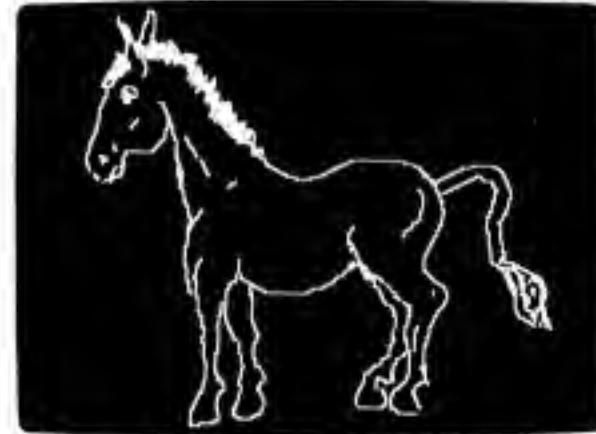
Once you've got a picture on the screen, you might want to edit or change small portions of it. For example, you've drawn this picture:



and you want to fix up the little "glitches" around the edge. There are a couple of ways to do this: you could set the PEN COLOR to black and DRAW the glitches out, you could erase whole portions of the screen and redraw them, or you could simply erase the whole thing and start over. Fortunately, there's an easier way. Press the pen to the square marked DOTS. You are now in DOTS mode, and will remain in DOTS mode until you specify another. While you are drawing with DOTS, the

Graphics Tablet will let you plot individual points on the screen. Each time you press the pen down in the working area you will plot one, and only one, point on the screen. When you lift the pen up again and press it down in a new place, you will plot another single dot. The dots will be of the color you specified in the most recent PEN COLOR command, or white if you have not selected any other color.

By setting the PEN COLOR to the BackGround COLOR (normally black), setting DOTS mode, centering the crosshairs on the extraneous glitches in the picture, and exorcising them one by one, you can turn a rough picture like the previous one into this:



DOTS mode is also handy for adding shading and texture to your pictures. DOTS mode is most useful when used with VIEWPORT and REDUCER, described later in this chapter.

To leave DOTS mode, press the pen in the command square for any other mode (like DRAW, LINES, BOX or FRAME).

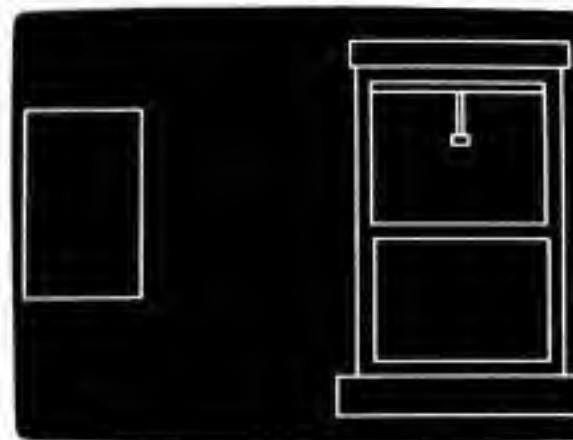
THE BIG FRAME-UP

LINE	DOT	FRAME	BOX	CATALOG	LOAD	SAVE	SEPARATE	SLIDE	AREA	DISTANCE
------	-----	-------	-----	---------	------	------	----------	-------	------	----------

When you're in the FRAME mode you can draw open rectangular boxes on the screen by specifying two diagonally opposite corner points. To enter FRAME mode, press the pen down in the (surprise!) FRAME command box. Now press the pen down anywhere on the Tablet's working area and lift it again. A single dot will appear on your screen. Take the pen and press it down at another point on the working area. The Apple will draw an open rectangle with opposite corners at the points you specified. Pick another point and press the pen down. Notice that the FRAME mode doesn't draw a frame with the new point and previous

point (as LINE mode would draw a line between them), but instead uses the new point as a corner of a separate FRAME. Pick and press a fourth point to complete the second FRAME.

Your FRAMES can be simple, or you can use many FRAMES to make a larger, more complicated FRAME:



You can draw your FRAMES in different colors, too. The FRAMES will be drawn in the current PEN COLOR, or white if you haven't selected any other color. Beware! Colored FRAMES may come out with a side or two missing because of the nature of the Apple's high-resolution graphics screen (see Appendix C). If this happens, re-draw the FRAME, but move the corner points very slightly to one side.

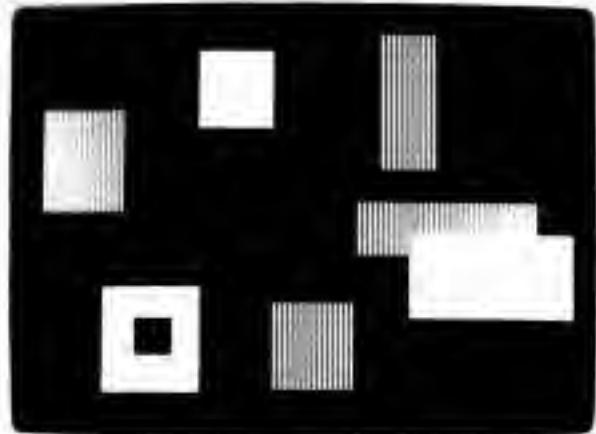
To leave FRAME mode, press the pen in the square for any other mode (such as DRAW, LINES, BOX or FRAME).

LITTLE BOXES

LINE	DOT	FRAME	BOX	CATALOG	LOAD	SAVE	SEPARATE	SLIDE	AREA	DISTANCE
------	-----	-------	-----	---------	------	------	----------	-------	------	----------

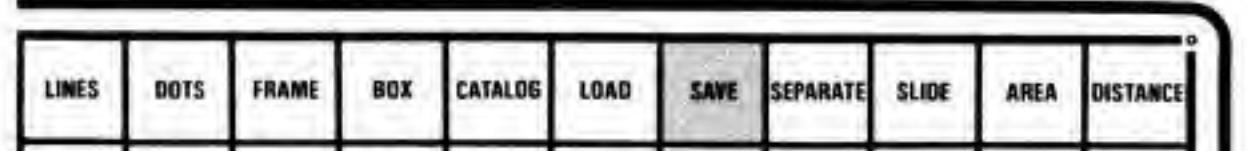
There's a white one, and a blue one,
And a green one, and an orange one,
And they're all made
On the Graphics Tablet
And they all look
Just the same.

Now FRAMES are nice, but they're kind of vapid. You might even go so far as to say they're empty. If you're looking for something a little more, well, fulfilling than an ordinary rectangular quadrilateral, then the BOX mode is for you. Press the pen down in the square marked BOX. Now press the pen down at two points on the working area, as you did for FRAME. The Tablet software will give you solid indication that the task is completed by drawing a uniform, monolithic box with corners at the two points you specified.



You will remain BOXed into this mode until you free yourself by pressing the pen down in one of the squares marked DRAW, LINES, DOTS, or FRAME. The BOXes you draw will be of the current PEN COLOR. If you have not specified a different color, your boxes will be white.

SAVING PICTURES FOR POSTERITY



By now you should have generated some beautiful (well, at least interesting) artwork. It's a shame that you have to erase it, isn't it? Well, you can save the entire picture for later recovery and further work by pressing the pen in the square marked SAVE. Your picture will vanish (temporarily) and the screen will display the message

TYPE A NAME FOR THIS PICTURE.

==>

A picture name can be from 1 to 26 characters long, and may include letters, numbers, and special characters (except the comma). Unlike normal diskette file names, picture names do not need to begin with a letter; you can have picture names such as

I FOR THE ROAD

or

<<SPACE>>

(notice the spaces before the name)

The reason for this is that before the Apple saves the screen onto the diskette, it adds the prefix "PIC." to your picture name to identify it as a bona fide Graphics Tablet Picture. Since diskette file names will always begin with the letter P (in PIC.), your picture names can begin with whatever you please.

The PIC. flag also implies that the picture file includes the Tablet WINDOW setting (see the WINDOW command). Files which do not contain this information should not carry the PIC. flag.

After you type the name of the picture, press **RETURN**. If you decide you don't really want to save the picture yet, just press **RETURN** without typing any name. Your picture will reappear, and you'll be left in DRAW mode.

If you do choose to save your picture, the Apple will then ask you:

DRIVE? ==> (DEFAULT=1)

The Apple will save your picture onto the diskette in the disk drive you indicate. The DEFAULT drive is the drive which the Apple thinks you'll want to use, drive 1 the first time and the drive specified previously each time thereafter. Type the drive number you wish to use and press **RETURN**, or just press **RETURN** to select the DEFAULT drive. (If you try to specify any drive number other than 1 or 2, the Apple will use the default drive). The drive will whirr and chug for a moment, then your freshly saved picture will reappear, in DRAW mode with PEN COLOR as it was when you left.

If there is already a picture on the selected diskette with the name you specified, the Apple will display the message

A PICTURE ALREADY EXISTS WITH THAT NAME.

CONTINUE (Y OR N)

If you wish to overwrite the current picture which has the name you specified, press **Y RETURN**. If you don't want to destroy the picture on the diskette, press **N RETURN** and repeat the SAVE operation using a different picture name (a lone **RETURN** is accepted as an **N RETURN** response).

If you complete the SAVE procedure, or if your attempt to SAVE a picture is foiled, and you get an error message from the Apple, you will lose any VIEWPORT you may have set (see the VIEWPORT command). If, however, you have aborted the SAVE command with an **N RETURN**, the VIEWPORT will remain intact.

If you receive this message:



then any number of things could be wrong: the diskette is full and can hold no more pictures, the diskette is write-protected, or there's another picture on the diskette with the same name and the file which holds that picture is locked. In the first case, simply use another uninitialized diskette. In the second case, remove the diskette, peel off the write protect tab and reinsert the diskette. In the third case, try another file name. Whatever the problem is, you may press the spacebar to attempt to SAVE the picture again under the same name, or press **RETURN** to cancel the attempted SAVE. Your picture will reappear, and you will be back in DRAW mode.



If you filled up the diskette by trying to SAVE a picture, only part of the picture will actually be stored on the diskette. It is best to delete the partial file from the diskette after you have SAVED the picture on another diskette (see **GETTING OUT**).

If you receive this message:



while attempting to SAVE a picture, then there are problems. Maybe you specified Drive 2 when you only have one drive, or the diskette is

uninitialized, or the data on it has been destroyed. You could have a faulty disk drive or controller card, or your drive is under the influence of a powerful magnetic field (did you put it on top of your television? Naughty, naughty). There may be no diskette in the drive, or you left the drive door open, or the diskette is crimped and is not rotating. Whatever the cause, press the spacebar to attempt the SAVE again, or press **RETURN** to get back to your picture. Investigate.

BROWSING THROUGH THE CATALOG

LINES	DOTS	FRAME	BOX	CATALOG	LOAD	SAVE	SEPARATE	SLIDE	AREA	DISTANCE
-------	------	-------	-----	---------	------	------	----------	-------	------	----------

You can look at the contents of your diskette by pressing the pen in the square marked **CATALOG**. The Apple will ask you (as above) for the drive number. You can select the **DEFAULT** drive by pressing **RETURN**.

The message

PRESS SPACE BAR TO CONTINUE

will be centered at the top of the screen. The drive will whirr a bit, and the names of all files on the diskette will be presented. Don't press the spacebar yet! The file names which begin in **PIC.** are your Graphics Tablet pictures, and should all have the annotation **B 033** to the left of them. Incidentally, the **B** denotes that they're **BINARY** files, and the **033** means that they use 33 diskette sectors, or a little over 8K bytes of memory, each. If you see any **PIC.** files which aren't marked **B 033**, then they're not complete pictures. Change their names (see **GETTING OUT**) so you don't mistake them for Graphics Tablet pictures in the future.

If you order a **CATALOG** of the Graphics Tablet Software diskette, these files will be included:



These are all component programs of the Graphics Tablet package. The only notable file is the **GRAPHICS TABLET LOGO**, which you'll notice has the notation **B 014** to the left of it. (The **014** means that it's

slightly larger than normal Tablet pictures, which are labeled B033. This causes no problems, however.) This is the picture of the Graphics Tablet Logo frame, which you see when you boot the diskette. You can LOAD this picture and work on it, even though it's not a PIC. file (see LOAD, below, for details).

CATALOG can fall victim to the same I/O ERROR problems as noted in SAVE. See the previous section for details.

If the CATALOG listing is too long for the screen, the listing will pause after displaying 18 files. Press the spacebar to get the rest of the CATALOG.

When you've finished looking at the CATALOG, just press the spacebar. Your picture will instantly reappear on the screen, with PEN COLOR unchanged.

GETTING LOADED

LINES	DOTS	FRAME	BOX	CATALOG	LOAD	SAVE	SEPARATE	SLIDE	AREA	DISTANCE	.
-------	------	-------	-----	---------	------	------	----------	-------	------	----------	---

Once you have SAVED a picture on diskette, you can call it back to your screen to be worked on some more, or just bring it out so you can admire it for a minute. Press the pen to the square marked LOAD. The following words will appear:

PLEASE TYPE THE PICTURE NAME.

If you change your mind and don't want to LOAD a new picture, press **RETURN**. Your previous picture will reappear, and you will be left in DRAW mode with the same PEN COLOR as when you left.

If you do want to LOAD another picture, type the name of the picture which you wish to see, and press **RETURN**. You don't have to type the PIC., the Apple will supply that for you. The rules for naming pictures are the same as described in the SAVE command.

You will then be asked to specify which drive the diskette with the chosen picture is in. Press **RETURN** to indicate that it's in the DEFAULT drive, or type the drive number (1 or 2) and press **RETURN**.

The disk drive will spin for a few moments, then the selected picture will appear on the screen. You will be left in DRAW mode.

LOAD is vulnerable to the same disk I/O ERROR problems as were described in the SAVE section.



You can LOAD picture files which were not created by the SAVE command, as long as they carry the notation B 034 in their CATALOG listing. One such file is the GRAPHICS TABLET LOGO file on your Graphics Tablet Software diskette. Even though this file doesn't have the PIC. flag in front of its name, it can be LOADED and worked upon like any other picture. If you LOAD this file, and SAVE it again, the new version will have the prefix PIC. attached to the name, and will have the notation B033 to its left in the CATALOG.

When the Apple sees the PIC. prefix, it infers that the file contains information about the Tablet WINDOW setting along with the picture. The absence of the PIC. flag indicates to the Apple that it should use the default WINDOW setting (see the WINDOW command). In addition, if you have a picture on the diskette whose file name does have the PIC. prefix, you can make the Apple ignore the Tablet WINDOW setting in that file by typing the PIC. prefix at the beginning of the file name when you LOAD it.



If you LOAD a picture which was SAVED on another Apple or Graphics Tablet, it's possible that the Tablet which created that picture uses a slightly different WINDOW setting than yours. The difference usually appears as a discrepancy between the motions of the pen across the working area and of the crosshairs on the screen. If the crosshairs don't correspond to the pen position, then re-LOAD the picture, but type the PIC. prefix at the beginning of the file name. This will make the Apple use the proper WINDOW setting for your Tablet.

With one exception, an attempt to LOAD a picture, whether successful or not, will remove any VIEWPORT you may have set (see the VIEWPORT command). If you have aborted the attempted LOAD with a **N RETURN** or a **RETURN**, this rule does not apply.

GETTING OUT



With the Graphics Tablet, you can perform three simple operations with disk files: SAVE, CATALOG, and LOAD. In order to RENAME or DELETE picture files, you'll have to leave the Graphics Tablet Software and get back to the Applesoft/DOS command level. To do this, get to DRAW mode and press **ESC**. The Apple will ask you if you indeed wish to leave.



If you answer **Y**, you will lose whatever picture you had on the screen! Any other reply will send you back to your artwork, in DRAW mode.

If you answer **Y**, then the Apple will run the HELLO program on the diskette, and you will see the Graphics Tablet Logo frame (see Chapter 1, STARTING UP).

Press **ESC** again to get to the HELLO menu. Choose **Q** to QUIT and press **RETURN**. The screen will be cleared and the Applesoft prompt character (**!**) will appear in the upper-left corner.

Now you can DELETE, RENAME, LOCK, UNLOCK, or VERIFY any of your picture files on the diskette, or do almost anything else in Applesoft or with DOS. (For details on how to perform these operations, see Chapters 2 and 4 of your DOS manual.) Remember to include the PIC. at the beginning of the pictures' file names! To return to the Graphics Tablet software, type

RUN HELLO

When the Graphics Tablet logo appears, press **ESC**, select **G** for Graphics Tablet Software, and press **RETURN**. You'll be working with the Tablet again, with a blank screen, a white pen, no VIEWPORT, the WINDOW at its default setting, the REDUCER off, and in DRAW mode.



DON'T try to RUN the file GRAPHICS TABLET SOFTWARE directly! It is not a program in itself, but is an EXEC file which runs several programs and sets up some parameters necessary for the well-being of the Tablet software. You should always enter the Graphics Tablet Software by selecting it from the HELLO menu.

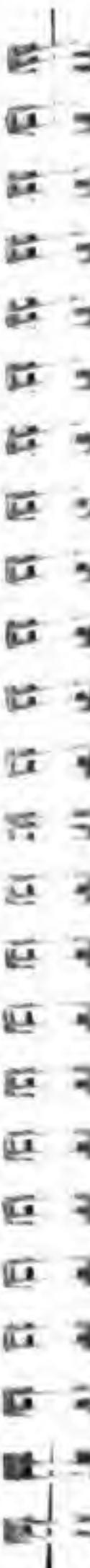
ROOM WITH A VIEWPORT

RESET	CLEAR	WINDOW	B6 COLOR	DELTA	SOFT RESET	VIEW PORT	CALIBRATE	REDUCER	PEN COLOR	DRAW
-------	-------	--------	----------	-------	------------	-----------	-----------	---------	-----------	------

Am I to spend the rest of my short life
Confined by these four corners, bright and sharp?
Shall I be limited in my designs
To draw only within this VIEWPORT small?

This cannot be! And yet, there's recompense:
The box which limits, also can protect,
And keep me from destroying what I've wrought.
Confine, protect; the VIEWPORT functions thus.

You can use the VIEWPORT command to select a rectangular area on the screen. Once you set an area for a VIEWPORT, you will be allowed to draw only within that area. This allows you to concentrate on one



area of the screen at a time, while protecting the rest of the picture from being accidentally erased or overdrawn.

The VIEWPORT appears on the screen as four small "L"-shaped corner marks, one at each corner of the VIEWPORT. Each leg of each "L" is three dots long and one dot high. The VIEWPORT itself is the area enclosed by these four corners (the corners are actually outside the VIEWPORT proper). If you change or remove a VIEWPORT, the corner marks vanish without a trace, leaving the screen under them unchanged.

To specify a VIEWPORT, press the pen to the VIEWPORT square on the Tablet menu. The prompting message

UPPER LEFT?

will appear briefly at the bottom of the screen. Position the pen at the spot where the upper-left corner of the VIEWPORT should be (imagine you're drawing a FRAME) and press down. One corner mark will appear, and another prompting message:

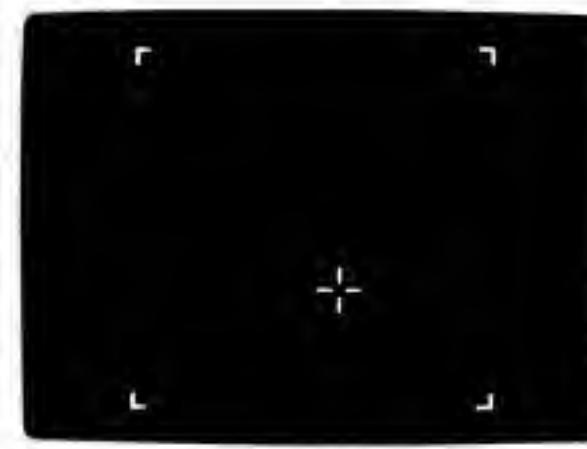
LOWER RIGHT?

will be displayed:



LOWER-RIGHT?

Position the pen at the opposite (lower-right) corner of your proposed VIEWPORT, and press down. The other three corners of the viewport will appear. This is how the finished VIEWPORT will look:



Unlike FRAME and BOX modes, in which you can specify the corner points in any order, VIEWPORT really does want the second corner point to be below and to the right of the first. If you give the points in reverse order, or specify an impossible VIEWPORT (one which has no height or width), then you will receive the message



You will then be asked for both corner points again.

Once you've set a VIEWPORT, what do you do with it? Simple, just DRAW. After you specify the two corner points, and you see the four-cornered frame, you will be placed in DRAW mode with the PEN COLOR unchanged. Anything you DRAW outside of the VIEWPORT simply will not show up on the screen; anything you draw inside it, will. Notice that the crosshairs will appear even outside the VIEWPORT, but pressing down on the pen has no effect.

You can change to any other drawing mode (BOX, DOTS, LINES, or FRAME) and it will work normally inside the VIEWPORT. But if, while you're in one of these modes, you try to specify a point outside the VIEWPORT, you'll receive the admonition

POINT OUTSIDE VIEWPORT. RESPECIFY.

Just choose another point inside the VIEWPORT. If you want to restart the BOX, FRAME, or LINE you're drawing, press the pen to the square for the proper mode again.

If you invoke the VIEWPORT command when another VIEWPORT is already active, the Apple will remove the previous VIEWPORT before asking you to specify a new one.

At any time after you have invoked the VIEWPORT command, but before you have finished specifying a new VIEWPORT, you can tell the Apple to give you one of two special VIEWPORTs. One of these is the VIEWPORT you were using before you started to set a new one, and the other is the "default" VIEWPORT (the currently set WINDOW).



To recover the VIEWPORT you had before you started to set a new one, press **RETURN** before you finish the VIEWPORT command.



To request the default VIEWPORT, press **D** before you finish the VIEWPORT command. The default VIEWPORT is the full screen, or (if you have invoked the WINDOW command) the area within the WINDOW. When the VIEWPORT is set to the full screen, no corner marks appear.



The VIEWPORT command always leaves you in DRAW mode with the PEN COLOR unchanged.



A BRILLIANT REDUCTION

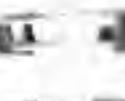
RESET	CLEAR	WINDOW	BG COLOR	DELTA	SOFT RESET	VIEW PORT	CALI-BRATE	REDUCER	PEN COLOR	DRAW
-------	-------	--------	----------	-------	------------	-----------	------------	---------	-----------	------

Once you've set a VIEWPORT, you can use the REDUCER function to shrink the entire Tablet working area into the VIEWPORT on the screen. This allows you to convert large pen motions on the Tablet into small motions on the screen. This lets you make precise, small drawings. When you use the REDUCER in conjunction with the DOTS mode, you can modify very small areas of a picture, setting and resetting individual dots if necessary.

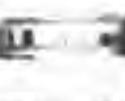
Once the REDUCER is enabled, it will stay in effect until you remove it or change the VIEWPORT. To use the REDUCER, set a VIEWPORT around the area in which you wish to work, then press the pen to the square marked REDUCER. When you hear the Apple beep, the REDUCER is active. If you receive the message



NOT POSSIBLE.



then you have specified a VIEWPORT which is too small or too disproportionately shaped for the REDUCER to function. Such an impossible reduction will leave you with the REDUCER inactive and everything else unchanged.



The screen position of the VIEWPORT determines the minimum possible size into which you can REDUCE. You can REDUCE into smaller VIEWPORTS in the upper-left corner of the screen than in the lower-right corner. Specifically, the smallest possible VIEWPORT size into which you can REDUCE ranges from two screen dots square (at the normal WINDOW setting) in the upper-left corner to 45 dots square in the lower-right.

To disable the REDUCER, press the pen to the REDUCER square again. The Apple will beep and the REDUCER will be disabled. The RESET, SOFT RESET, WINDOW, VIEWPORT, and LOAD commands also disable the REDUCER.

When the VIEWPORT is at its default setting, the REDUCER has no effect.

OPENING THE WINDOW



The WINDOW command works like a VIEWPORT with the REDUCER on, but the other way 'round. Where VIEWPORT with REDUCER lets you draw something large on the Tablet, and have it appear smaller and in a specific place on the screen, the WINDOW lets you draw something small in a specific place on the Tablet and have it appear large on the screen.

You normally set a WINDOW before you begin drawing a picture. Press the pen to CLEAR and then to WINDOW. This will appear on the screen:

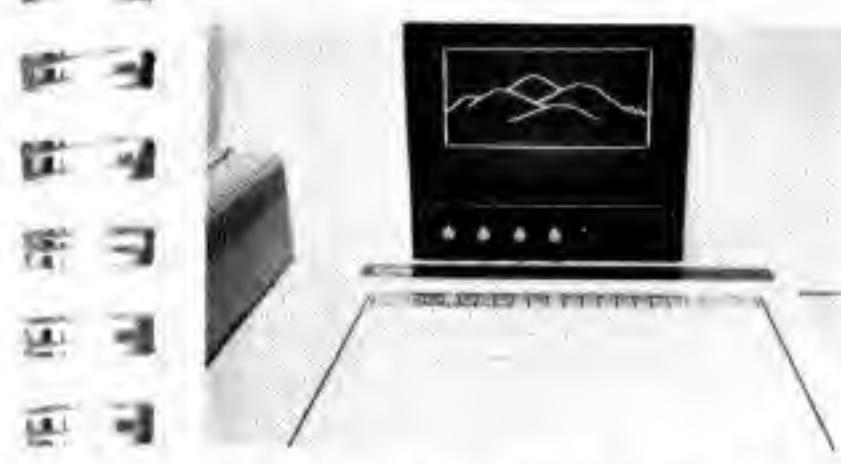


Find a picture of a molehill and tape it to the Tablet's working area. Now take a pen or a pencil (not the Tablet's pen!) and draw a box around the significant part of the picture. Take the Tablet's pen and, following the highlighted instructions on your screen, press it to the upper-left corner of the box. The highlighting will shift:



PRESS PEN AT
UPPER-LEFT AND LOWER-RIGHT
CORNERS OF THE ACTIVE
TABLET AREA DESIRED

Press the pen to the lower-right corner. The words will disappear and the drawing screen will return, with a large frame in the middle of the screen. This frame is proportional to and corresponds with the frame around the molehill on the Tablet, and is centered on the screen. Take the pen, set DRAW mode, and trace the molehill. You will make a mountain on your screen out of the molehill taped to the Tablet.



After it's finished, WINDOW returns you to DRAW mode, with your PEN COLOR unchanged and the VIEWPORT set to the same size as the WINDOW frame on the screen.

The reason that you aren't shown the screen and crosshairs while you set the WINDOW (as you are when you set a VIEWPORT) is that you're selecting an area on the Tablet, not the screen. The resulting area on the screen is as large as the Apple can make it, proportional in size to the WINDOW on the tablet, and centered on the screen. Since the WINDOW area on the Tablet bears little relation to the screen before it's set, the screen and crosshairs are not displayed.

At any time after you have initiated the WINDOW command and before you have completed it, you can use the Apple's keyboard to indicate that you want the default WINDOW, (the entire working area of the Tablet, !

or that you want to cancel the WINDOW sequence. Press **RETURN** at any time during the WINDOW sequence to cancel it; press **D** to select the default WINDOW.

BROKEN WINDOWS

If you receive the message

PLEASE SPECIFY POINTS CORRECTLY!

then you've not specified the two corner points in their proper upper-left, lower-right order, or you've tried to set the WINDOW to an area on the Tablet that's too small. You will be asked to specify both corners again. If you want to cancel the attempted WINDOW, press **RETURN**.

If the Apple flashes the message

PLEASE STAY WITHIN THE WORK-AREA.

then you've let the pen stray outside the working area of the Tablet's overlay. You will be prompted again to indicate the corner point. To cancel the WINDOW command, press **RETURN**.

DRAWING IN THE WINDOW

Once you've placed a WINDOW on the Tablet, you can use any of the Tablet's drawing modes (DRAW, LINES, DOTS, FRAME, or BOX) to draw, as long as you stay within that WINDOW.

You can set the VIEWPORT within the WINDOW on the screen. Once you've set it, you can even REDUCE into it, and use the entire Tablet area within the VIEWPORT. When you turn the REDUCER off, you will again be limited to your WINDOW.

Once you've set a WINDOW, the only way to remove the WINDOW frame is to set a new WINDOW or use the Tablet RESET command. No other Tablet command will remove a WINDOW. The REDUCER will allow you to temporarily override the WINDOW; when you turn off the REDUCER, you will be left with the previous WINDOW again. Experiment with using WINDOW and the REDUCER; you'll be surprised at what they can do.



When you specify a WINDOW on the Tablet, the Apple will draw the WINDOW frame on the screen on top of the current picture. The sides of the frame are two dots wide, and the top and bottom are one dot wide. If you set a new WINDOW, the Apple will remove the frame by drawing over it with the BackGround COLOR. The WINDOW command can

therefore destroy parts of your previous picture. Also, even though WINDOW sets the VIEWPORT to the portion of the screen inside the WINDOW frame, the CLEAR command will clear the entire screen, including everything outside the VIEWPORT and the WINDOW frame, and even the frame itself! (The frame will be redrawn after the CLEAR.) If you reset the VIEWPORT to a slightly smaller size than the WINDOW, the CLEAR command will work normally and erase only what is within the VIEWPORT.

The WINDOW information is stored along with the picture information when you SAVE a picture onto diskette. There is no way to avoid saving this information. When you LOAD a picture, the Apple will automatically use the WINDOW setting stored with that picture, if that picture's diskette file name begins with PIC. If it does not, the Apple will use the default WINDOW setting for your Tablet. You can force the Apple to ignore the WINDOW setting stored in a picture file in three ways:

1) Leave the Graphics Tablet Software (see GETTING OUT) and RENAME the file, removing the PIC. prefix from the file name.

-- or --

2) When you LOAD the picture, type the PIC. flag at the beginning of the picture name. (LOAD PIC.filename)

-- or --

3) Once you have LOADED the picture, press the pen to WINDOW and type **D** to get the default setting.

RESET

RESET	CLEAR	WINDOW	BG COLOR	DELTA	SOFT RESET	VIEW PORT	CALIBRATE	REDUCER	PEN COLOR	DRAW
■	■	■	■	■	■	■	■	■	■	■

The RESET command lets you "wipe the slate clean" and begin anew on a fresh picture. Namely, it:

1) Sets the WINDOW to the the normal 11 inch wide, 6.5 inch tall rectangle at the top of the working area.

2) Sets the VIEWPORT to the full screen.

3) Sets CALIBRATE to one unit per screen dot, and leaves the unit type undefined.

4) Sets the BackGround COLOR to black, and clears the screen.

5) Sets the PEN COLOR to white and sets DRAW mode.

- 6) Sets the value of DELTA to 2 and turns the Audio Feedback feature OFF.
- 7) Sets the default drive number for LOAD, SAVE, and CATALOG to 1.

Using the RESET command is just like restarting the Graphics Tablet software all over again.

A SOFTER RESET

RESET	CLEAR	WINDOW	BG COLOR	DELTA	SOFT RESET	VIEW PORT	CALIBRATE	REDUCER	PEN COLOR	DRAW
-------	-------	--------	----------	-------	------------	-----------	-----------	---------	-----------	------

The SOFT RESET command is a milder version of RESET. It lets you reset many of your drawing and calculating functions, while leaving your picture, WINDOW settings, and pen color intact. SOFT RESET:

- 1) Sets the VIEWPORT to the full screen, or to the currently set WINDOW. This is the same as pressing **D** while setting a VIEWPORT.
- 2) Sets the CALIBRATE unit to 1 and the unit type undefined.
- 3) Sets the DELTA value to 2 and turns the Audio Feedback feature OFF.
- 4) Sets DRAW mode.

Nothing else is changed by SOFT RESET. The PEN COLOR, BackGround COLOR, the WINDOW setting, the default drive number, and so on, all remain the same.

CALIBRATE

RESET	CLEAR	WINDOW	BG COLOR	DELTA	SOFT RESET	VIEW PORT	CALIBRATE	REDUCER	PEN COLOR	DRAW
-------	-------	--------	----------	-------	------------	-----------	-----------	---------	-----------	------

The CALIBRATE command lets you specify a distance on the Tablet surface and use it for measuring with the DISTANCE and AREA functions.

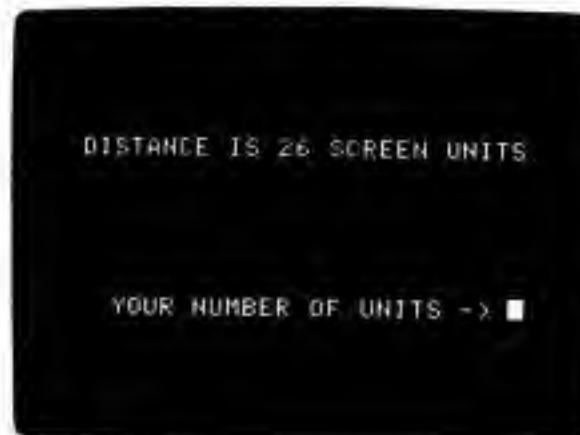
Press the pen to the CALIBRATE square on the menu. The Apple will beep and the brief question:

BEGINNING POINT?

will appear at the bottom of the screen. Select a point and press down. Another brief question:

ENDING POINT?

will flash at the bottom of the screen. Select another point, say, an inch away from the first and press down. The screen will vanish and the following frame will appear:



The Apple has converted the distance between the two points you specified into its internal "screen units". You now have the opportunity to define how long that distance actually was. If you've just arrived from Alpha Centauri, and you specified a distance of about one U.S. inch, then that's about 5 glibbets. Type

5 **RETURN**

G L I B B E T S RETURN

You've now defined the length you specified to be 5 Alpha Centauri glibbets. You're free to change it, of course, and give a distance of 10 chronacs, 200 malms, or even half a gretton if you so desire (use decimal numbers for fractions, in this case **█ 5** grettons).

The CALIBRATE command won't let you specify negative distances, or distances greater than 999999999. It also won't let you use a name for your measurement which is more than 10 letters, numbers, or special characters (such as asterisks, bracketts, etc.) long.

You can tell the Apple that you'll agree to use its internal screen units for measurement by answering its questions about length and name with the **RETURN** key. Once you've specified a distance and unit name in the CALIBRATE command, they will remain as you set them until you: a) reset them with the CALIBRATE command, b) do a RESET or a SOFT RESET, or c) change the BackGround COLOR, the VIEWPORT setting, or the WINDOW setting.

MISTAKEN CALIBRATION

If you specify an endpoint for the CALIBRATE distance which is outside the current VIEWPORT, you will be asked to indicate the point again.

Once you begin to define a distance for the CALIBRATE command, you can cancel the procedure by pressing the **RETURN** key.



Don't change the REDUCER setting after you've CALIBRATED your Tablet. If you do, it will shrink your measurements just as it shrunk your Tablet movements, and all your DISTANCE and AREA calculations will be incorrect.

LONG DISTANCE...

LINES	DOTS	FRAME	BOX	CATALOG	LOAD	SAVE	SEPARATE	SLIDE	AREA	DISTANCE
-------	------	-------	-----	---------	------	------	----------	-------	------	----------

Once you've set a distance and a unit name with the CALIBRATE command, you can use those definitions to calculate the DISTANCE that you move the pen along a path on the Tablet surface.

For example, find a road map (we'll use one of Central California), unfold it, and tape it to the Tablet so that the legend (with the scale of distance) is in the working area. Use the CALIBRATE command to set the distance and unit name to the scale of distance on the map:



Now point the pen to the DISTANCE square and press down. The Apple will beep, signaling you to take the pen and trace a path on the map. Draw a path from Buttonwillow to Bakersfield, along Route 5. The path



will appear on the screen as you draw. When you lift the pen, the Apple will beep again and flash

CALCULATING...

at the bottom of the screen, and then go away and think for a moment. It will soon return, telling you that the DISTANCE you traveled from Buttonwillow to Bakersfield is about 25 miles. After a short delay, you will be returned to DRAW mode.

The path you draw, as it appears on the screen, is just like any other path in DRAW mode, and it is subject to the same DELTA effects as DRAW. Lower DELTA settings will give you more accurate DISTANCES; higher DELTA settings will give you less accurate (albeit quicker) approximations.

If you invoke the DISTANCE command and then decide you don't want to calculate a distance after all, simply press the **RETURN** key instead of drawing a path on the Tablet.

There is a limitation on the maximum DISTANCE your path can be. The longest path you can draw for DISTANCE contains 800 points. With a DELTA setting of 2, this is 1600 screen units, or about 59 actual inches on the Tablet. Of course, this will be different if you're using the WINDOW or REDUCER functions.

... AND AREA CODES

LINES	DOTS	FRAME	BOX	CATALOG	LOAD	SAVE	SEPARATE	SLIDE	AREA	DISTANCE
-------	------	-------	-----	---------	------	------	----------	-------	------	----------

The AREA command is a counterpart to the DISTANCE command. But instead of letting you find the DISTANCE between Buttonwillow and Bakersfield, CA, it will let you figure the AREA of Manhattan. Quickly remove the map of Central California and switch to one of the New York City area. Use the CALIBRATE command with the scale of distance on the new map.

Now place the pen on the AREA square and press down. The Apple will respond with a beep. Trace the perimeter of Manhattan. As soon as you lift the pen, the Apple will beep again and flash

CALCULATING...

and sit and think for a few moments. Soon it will return with the area of the island, expressed in the units you set in the CALIBRATE command. It will hold this value on the screen for about five seconds, and then return you to DRAW mode.

Now CLEAR the screen and try it again. This time, don't go completely around the island, but stop about half an inch away from your starting point. The Apple will obligingly close the curve for you, connecting the ending point directly to the beginning, before it calculates the AREA.

AREA is subject to the same limitations as DISTANCE: you can only draw a path 800 points long, or about 59 Tablet inches with a DELTA setting of 2. As in DISTANCE, a larger DELTA setting will give you less accurate results. And if you had the REDUCER on when you CALIBRATED, don't turn it off when you are going to calculate an AREA. A press of the **RETURN** key will abort the AREA command, just as it will for the DISTANCE command.

Now CLEAR the screen and find the AREA of Manhattan again. This time, go around the island twice. Notice that even though the AREA looks the same on the screen, the number that the Apple will return is about twice the actual AREA of the figure. This is normal: if you go around three times, the Apple will give you a number three times too large, and so on.

If, while drawing around an AREA, you move the pen outside the VIEWPORT, the Apple will act as if you had lifted the pen at that point, close the curve, and figure the AREA.

SLIDE RULES

LINES	DOTS	FRAME	BOX	CATALOG	LOAD	SAVE	SEPARATE	SLIDE	AREA	DISTANCE
-------	------	-------	-----	---------	------	------	----------	-------	------	----------

Once you've got a picture on the screen, you don't have to rest at that. No, you can mobilize your pictures, give them some motivation, see some action! Press the pen to the SLIDE square. The request BEGINNING POINT?

will appear briefly at the bottom of the screen. Use the pen to select any point on your picture, and press down. A second request

ENDING POINT?

will appear. Select another point on the screen, some distance removed from the first. Watch your picture travel across the screen, both vertically and horizontally, until the first point you selected (on the picture) is in the vicinity of the second point (on the screen).

If you decide that a SLIDE isn't what you want right now, press **RETURN** to cancel the operation. You'll be returned to your picture, in DRAW mode.



The SLIDE operation is performed in four directions, with what mathematicians call "toroidal wrap-around". This ponderous phrase means that the picture thinks it's not on a flat screen, but wrapped around a doughnut: the left side is joined to the right side, and the top is joined to the bottom, so that everything that you SLIDE off one edge of the screen will reappear on the opposite edge. When your SLIDE is complete, you will be returned to DRAW mode.

SLIDE moves the entire screen: there is no way to move only a portion of the screen. Because of the way the Apple places colors on the screen (see A SHORT DIGRESSION...), the SLIDE command can move the picture the exact vertical distance you indicate, but can only come within 14 dots of the horizontal location you specify.

SLIDE will remove the VIEWPORT and WINDOW borders before it moves the picture, but will replace them in their former locations (not SLID over) after the SLIDE is complete.

PRISMATIC APPLE

LINES	DOTS	FRAME	BOX	CATALOG	LOAD	SAVE	SEPARATE	SLIDE	AREA	DISTANCE
-------	------	-------	-----	---------	------	------	----------	-------	------	----------

The SEPARATE function "strips" your picture, until only one color is left. Press the pen to the SEPARATE square. You will be presented with a color menu, just like in PEN or BackGround COLOR. REMEMBER: The SEPARATE command will destroy parts of your picture. If you want to preserve a picture, be sure to SAVE it before you do a SEPARATE. If you've already started a SEPARATE command, just press **RETURN** to cancel it and return you to DRAW mode.

If you do want to SEPARATE out your picture, select a color from the color menu with the pen and press down. The menu will vanish and your picture will reappear. Quicker than you can pronounce "refraction", your picture will be stripped of all colors except the one you selected. You will be left in DRAW mode, with your BackGround COLOR set to black and your PEN COLOR set to the SEPARATE color you specified.

There is no way to undo a SEPARATION. The SEPARATE command will remove any VIEWPORT or WINDOW before it performs its function, and restore them when it's finished. SEPARATE works only on the entire screen: there is no way to SEPARATE only a portion of the screen.

You cannot SEPARATE out the color black. If you did, you'd be left with a blank screen! The Apple will deny your attempt to separate out either of the blacks with the message

NO SEPARATION ON BLACKS.



IN CONCLUSION

Congratulations! If you've come this far, and practiced with your Tablet along the way, then you've mastered the basic functions of the Apple Graphics Tablet. With a little practice, you can be drawing and manipulating pictures with skill and ease. If you're interested in doing more with your Tablet, and you're accustomed to programming in Applesoft BASIC, then you might be interested in looking into Chapter 3. It describes the internal operation of the Graphics Tablet software, and the operating subroutines in the Graphics Tablet itself. You'll find dozens of new applications for your Tablet. Go ahead, keep drawing, and have fun!

CHAPTER 3 PROGRAMMING THE GRAPHICS TABLET

46	The Programs
47	Tablet-Code Applesoft
49	The Main Loop
50	Menu Items
51	Extending the Menu
51	Example: Instant Color Menu
53	Example: Circle Mode
55	Example: Disc Mode
56	The Firmware
57	Tablet Control
60	Accepting Input
61	From Machine Language
64	Quick-Draw
65	By Any Other Name

THE PROGRAMS

There are four main programs which comprise the Graphics Tablet software. Three of these programs are supplied on your GRAPHICS TABLET SOFTWARE diskette, and the fourth is stored in ROM (Read-Only Memory) on the Tablet Interface card. These programs are:

- 1) TABLET-CODE APPLESOFT: This is a large applications program, written in the Applesoft II BASIC programming language. This is the program which performs all the commands and functions of the Tablet as described in Chapter 2.
- 2) QUICK-DRAW: This is a machine language subroutine which is used by the TABLET-CODE APPLESOFT program. This subroutine allows an Applesoft program to draw lines on the Apple's high-resolution graphics screen as fast as the Tablet can supply the points. This machine-language subroutine is hidden inside an Applesoft program.
- 3) Tablet Firmware: This is a set of subroutines permanently stored in ROM on the Tablet's Interface card. These are base-level subroutines for the basic operation of the Tablet. They can be used from any Apple programming language.
- 4) UTILITIES: This is a package of machine-language subroutines which perform many of the screen manipulation functions of the TABLET-CODE program. This package includes the subroutines which perform the SEPARATE and SLIDE operations. It also includes the shape table used by the Applesoft DRAW command to draw the corner marks for the VIEWPORT. These subroutines are stored in a binary file on the diskette and load at location \$6000 (decimal 24576) in memory. The length of this file is \$330 (816 decimal) bytes.

In addition, there are two other Applesoft programs which are used as part of the Graphics Tablet software package, but don't take part in the actual operation of the Tablet. They are:

- 1) HELLO: This is the program which is executed when you boot the diskette. It is also executed whenever you exit the TABLET-CODE or MENU ALIGNMENT programs. It allows you to select which program you wish to run, and lets you quit if you want to.
- 2) MENU ALIGNMENT: This is another Applesoft program that sets up an information file on the diskette, called TAB.INFORMATION. This file contains information about what slot the Interface card is in and where the overlay is located on the Tablet.

TABLET-CODE APPLESOFT

This is the main operating program for the Graphics Tablet. It is written in Applesoft, and takes up 12K bytes of the Apple's memory. It resides between locations \$1000 and \$3FFF (decimal 4096 and 16383) of memory. It requires that your Apple have the Applesoft language in ROM or on a Language System Language card. It will not run with cassette or diskette versions of Applesoft.

A source listing of this program appears in Appendix D, along with an atlas of subroutines, variables, and special locations. Here is a brief map to the program:

Lines	Function
10-160	Initialization. This section reads the Tablet information file, sets up all pertinent Tablet parameters, and initializes and clears Page 2 of the Apple's high-resolution graphics screen. It also places the program and its variables in the proper locations in the Apple's memory, and loads the UTILITIES subroutines.
170-180	This is the main DRAW mode loop. These two lines take input from the Tablet pen and send it to the QUICK-DRAW subroutine to draw on the screen. The only way out of this loop is to press a key or press the pen outside the Tablet's working area (i.e., on the menu).
190-194	These lines are executed when you press a key during DRAW mode. If you press any key other than ESC , nothing happens. If you press ESC , you will be asked whether you wish to quit or not. Pressing any key other than Y will return you to DRAW mode. Otherwise, the HELLO program will be run.
200-290	These lines sense when you press the pen to the menu. Line 280 is the main menu vector table.
300-310	The CLEAR command.
330-410	The LOAD command.
420-520	The SAVE command.
530-540	A subroutine to input the disk drive number during LOAD, SAVE, and CATALOG.
550-560	The SOFT RESET command.
570-610	The CATALOG command.

620-640	The BackGround COLOR command.	1990-2030	The calculation section for AREA.
650	The PEN COLOR command.	2070-2080	The DISTANCE command. This section is the drawing loop.
660	This line lets you reenter LINES, DOTS, BOX, or FRAME mode after a menu selection.	2090-2120	The calculation section for DISTANCE.
670-870	This subroutine draws the color menu for BackGround COLOR, PEN COLOR, and SEPARATE, and lets you select a color with the pen.	2160-2290	The CALIBRATE command.
880	Color box low-resolution draw.	2300	A subroutine to blank out the four lines at the bottom of Page 2 of Text mode.
890-1120	The WINDOW command.	2310	A subroutine to display "BEGINNING POINT?" at the bottom of Page 2 of Text mode and wait for the pen to be pressed down.
1130-1140	This subroutine resets the Tablet WINDOW information after a color menu selection.	2320	A subroutine to display "ENDING POINT?" at the bottom of Page 2 of Text mode and wait for the pen to be pressed down.
1150-1290	The VIEWPORT command.	2330-2480	The SLIDE command.
1300	This subroutine causes a 1.15 second delay. It is used to pause while the Apple is displaying a message on the screen.	2490-2580	The SEPARATE command.
1310-1320	This subroutine waits until either a key is pressed on the keyboard or the pen is pressed down, and then returns.	2590-2600	This subroutine resets the Tablet firmware.
1330-1340	This subroutine draws or undraws the four corner marks for a VIEWPORT.	2610-2730	Error handling subroutines.
1350-1360	This subroutine draws a single VIEWPORT corner mark.		
1380-1390	The REDUCER command.		
1400-1440	Turns on the REDUCER.		
1460-1560	The DELTA command.		
1580-1660	LINES mode.		
1680-1720	DOTS mode.		
1740-1820	FRAME mode.		
1840-1930	BOX mode.		
1940-1950	This subroutine is called whenever you specify a point outside of the VIEWPORT for any of the four modes mentioned above.		
1970-1980	The AREA command. This section is the drawing loop.		
		1990-2030	The calculation section for AREA.
		2070-2080	The DISTANCE command. This section is the drawing loop.
		2090-2120	The calculation section for DISTANCE.
		2160-2290	The CALIBRATE command.
		2300	A subroutine to blank out the four lines at the bottom of Page 2 of Text mode.
		2310	A subroutine to display "BEGINNING POINT?" at the bottom of Page 2 of Text mode and wait for the pen to be pressed down.
		2320	A subroutine to display "ENDING POINT?" at the bottom of Page 2 of Text mode and wait for the pen to be pressed down.
		2330-2480	The SLIDE command.
		2490-2580	The SEPARATE command.
		2590-2600	This subroutine resets the Tablet firmware.
		2610-2730	Error handling subroutines.

THE MAIN LOOP

The main programming loop of the TABLET-CODE program occurs in lines 170 through 290. Lines 170 and 180 are the main loop for the DRAW mode. The DRAWing is done by the CALL EP% in line 170. This activates the QUICK-DRAW subroutine, which reads the Tablet and draws on the screen. As it draws on the screen, it also places the coordinates for each point plotted into the two arrays called X% and Y%. It uses the variable N% as an index into these arrays, and uses the value of the variable D% as its DELTA value (see the section on the QUICK-DRAW subroutine). The QUICK-DRAW subroutine returns to the Applesoft program when one of four events occur:

- 1) A key on the keyboard was pressed.
- 2) The pen was pressed down outside the working area of the Tablet.
- 3) The pen was lifted after being pressed down.
- 4) One of the arrays X% or Y% was filled up.

When QUICK-DRAW terminates, the termination condition code (a number from 0 to 3) is stored in location 700, and the Applesoft program

places this value into the variable CD. Depending upon the termination condition, the program either reinitializes the DRAW mode, or branches to line 190 (to handle a keypress) or line 200 (to get a menu selection).

If the pen was pressed outside the working area, then lines 220 through 240 sense the pen's position again in relation to the menu. Two numbers are returned: X holds a number from 0 to 21 which corresponds to one of 22 horizontal positions across the menu, and Y holds a number (0 or 1) which corresponds to one of two menu rows.

The ON Y+1 GOTO 280,290 statement in line 250 selects between the top and bottom menu rows. The ON X+1 GOTO 140... in line 280 selects among the 22 menu items in the top row.

MENU ITEMS

Each menu item corresponds to a block of code (not a subroutine) in the program. After the code for each item performs its function, it executes either a GOTO 170 (to reinstate DRAW mode) or a GOTO 660 (to reenter the current mode). A list of the variables which are used in the program and a description of their function appears in Appendix B. Subroutines which are called by parts of the program also appear in that Appendix.

The four other modes (LINES, DOTS, BOX, and FRAME) are actually independent menu items which operate differently from DRAW mode. For an example, look at the code for the FRAME mode, in lines 1740 through 1820. Line 1740 resets the Tablet to accept points from the current WINDOW area, with a sparkling crosshairs cursor displayed. It also sets the variable CM (for Current Mode) to 3, for FRAME mode. Line 1750 loops until the pen is pressed onto the Tablet surface, and returns the coordinates of the point in X and Y. These coordinates can be used directly to plot onto the high-resolution screen.

Line 1760 checks to see if the point is inside the current VIEWPORT. If it is not, the subroutine at line 1940 is called. This subroutine checks that the point is in the menu area. If it is, it sets the variable RT to 1; if not, it sets RT to 0 and displays the message

POINT OUTSIDE VIEWPORT. RESPECIFY.

Back in line 1760, if a menu item was selected, the code causes a jump to line 220 (menu selection). If the point was outside the Viewport, the mode is restarted by a GOTO 1750.

If the point was inside the VIEWPORT, then line 1780 plots the point on the screen and saves its coordinates in the variables TX and TY. Lines 1790 through 1810 go through the whole get-a-point procedure again, and get another point in X and Y. Finally, line 1820 actually draws the FRAME, and jumps back to line 1750 to get another two points.



EXTENDING THE MENU

If you can write programs for the Apple, then you can tailor the TABLET-CODE APPLESOFT program to your own liking. You can add extra functions and remove or modify existing functions. You can define your own menu selections, or you can even start from scratch and write your own programs to use the Graphics Tablet to do just about anything.



The TABLET-CODE APPLESOFT program uses almost all of the memory space allotted for it. If you wish to add a function to the code, you must delete some of the program to make room for it. If the program grows any larger, it will not work.



EXAMPLE: INSTANT COLOR MENU

If you're tired of having to wait for the Apple to redraw the color menu when all you want to do is change the PEN COLOR from white to black, here's a modification you can make to get instant changes in PEN COLOR. To do it, you'll have to sacrifice one of the Tablet's other functions. Since this will mean changing your TABLET-CODE program, it's important that you not work on the original backup diskette.



Type



NEW

LOAD TABLET-CODE APPLESOFT



to load the unmodified program. To make room for your new code, delete a function you don't use much (some good candidates for oblivion are SLIDE, SEPARATE, AREA, DISTANCE, and CALIBRATE — they are special-purpose functions and their removal won't affect the rest of the program). Let's delete the SLIDE function. Type

DEL 2330,2480



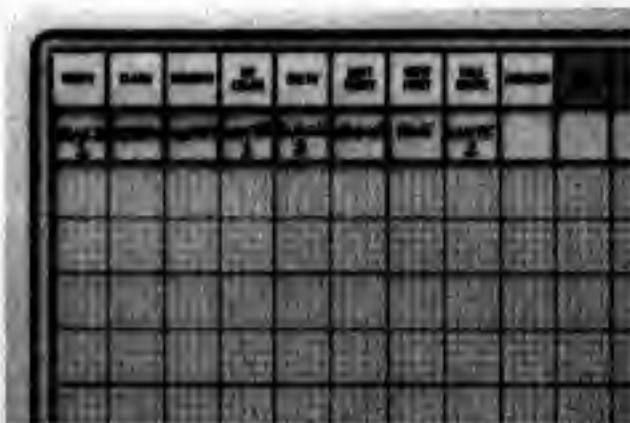
Now you've got about 250 more bytes to use for your program. To replace the SLIDE command with a null command, enter the line

2330 GOSUB 1130: GOTO 170



to reset the Tablet and return to DRAW mode if you try to select the now-defunct SLIDE function.

What we'll do is let you choose a new PEN COLOR simply by poking one of the first eight squares in the second row of the menu. Label these squares on your menu with a pencil or pen:



Now for the programming part. In lines 250 through 280, the variables X and Y hold the coordinates of the menu box which was just selected. Y is set to 0 for the top row and 1 for the bottom, and X holds a number between 0 and 21 corresponding to the 22 boxes in each row. So, if you poke the pen to one of the eight boxes of the new color menu, Y will be set to 1 and X will be a number from 0 to 7, depending upon which of the 8 squares you poked. It just so happens that the eight colors in Applesoft's high-resolution graphics mode are numbered 0 through 7, and they are in the exact same order as the color names you wrote on the second row of the menu! Isn't that lucky? But, first we've got to handle Y. In line 250, if Y is equal to 1, the program goes to line 290, the null function. Let's replace that line with

```
290 IF X<0 OR X>7 THEN GOSUB 1130: GOTO 170
```

Now the null function is executed only if the pen was pressed in the second row and not in the first eight boxes. If the pen was pressed in one of the first eight boxes, the next line will be executed. So let's make the next line:

```
295 PC=X: HCOLOR=PC: GOSUB 1130: GOTO 660
```

This line sets the Pen Color to the value of X (remember, 0 through 7?) and sets the high-resolution COLOR to that value. Then it resets the Tablet and goes to line 660, which reenters the current mode.

And that's it! Now before you RUN it save this version onto a copy of the GRAPHICS TABLET SOFTWARE diskette. You might want to add a REM statement at the beginning, describing the change and date. When you SAVE the program, you must save it under the name TABLET-CODE APPLESOFT or you won't be able to use it. Since the version of TABLET-CODE APPLESOFT that is on your diskette is LOCKed, you will have to UNLOCK it before you can save the new version.

To use your newly modified program, type

RUN HELLO

press **ESC**, select **G**, and press **RETURN**. When the program is running, you can instantly change colors in midstream, from any mode, by pressing the pen to the box for that color.

EXAMPLE: CIRCLE MODE

Here's another change which is a little more extensive than the previous one. Here are two extra modes, counterparts to BOX and FRAME, which draw open and filled circles rather than rectangles. You'll specify the location and size of a circle by poking two points on the Tablet: the first one will be the center of the circle and the second will be on the perimeter. Since this will require some extra room, (if you haven't done it already) delete a function such as SLIDE. In fact, you can delete both SLIDE and SEPARATE and use their menu squares to set CIRCLE and DISC mode. On a copy (not either of your original copies) of the GRAPHICS TABLET SOFTWARE diskette, type

```
LOAD TABLET-CODE APPLESOFT  
DEL 2330,2580
```

Now they're gone. Since CIRCLE and DISC need two points, just like BOX and FRAME, we can follow the example of those modes. Type

```
2330 REM ** CIRCLE MODE **  
2340 GOSUB 1130: PRINT D$;"IN#";SL: CM=5  
2350 RT=2: INPUT X,Y,Z: IF Z<>2 THEN POKE -16368,0: GOTO 2350  
2360 IF X<X3 OR X>X4 OR Y<Y3 OR Y>Y4 THEN GOSUB 1940: IF  
RT=1 THEN 220  
2370 IF RT=0 THEN 2350  
2380 HPLOT X,Y: TX=X: TY=Y  
2390 RT=2: INPUT X,Y,Z: IF Z<>2 THEN POKE -16368,0: GOTO 2390  
2400 IF X<X3 OR X>X4 OR Y<Y3 OR Y>Y4 THEN GOSUB 1940: IF RT=1 THEN 220  
2410 IF RT=0 THEN 2390
```

At this point, the coordinates of the center of the circle are in TX and TY, and the coordinates of a point on the perimeter are in X and Y. Let's find the radius of the circle now:

```
2420 R=SQR((X-TX)^2+(Y-TY)^2)
```

With a little trigonometry, we know that the horizontal and vertical distance from the center of a circle to any point on the perimeter is given by the simple formulae

2440 DX=R*SIN(TH): DY=R*COS(TH): X=TX: Y=TY

where R is the radius (derived in line 2420) and TH is an angle from 0 to 2π . Furthermore, we know that this formula gives us not one, but eight points on the circle:

X+DX, Y+DY	X+DX, Y-DY	X-DX, Y+DY	X-DX, Y-DY
X+DY, Y+DX	X+DY, Y-DX	X-DY, Y+DX	X-DY, Y-DX

as TH ranges from 0 to $\pi/4$ where X,Y is the center of the circle. So, let's add a loop and the lines to plot the points on the perimeter.

```
2430 FOR TH=0 TO .7854 STEP 1/R
2450 HPLOT X+DX,Y+DY: HPLOT X+DX,Y-DY: HPLOT X-DX,Y+DY:
  HPLOT X-DX,Y-DY
2460 HPLOT X+DY,Y+DX: HPLOT X+DY,Y-DX: HPLOT X-DY,Y+DX:
  HPLOT X-DY,Y-DX
2470 NEXT Th: GOTO 2350
```

Now to finish it all up, change the lines

```
660 ON CM+1 GOTO 170,1580,1680,1740,1840,2330
1380 RD=RD+1: IF RD>1 THEN RD=0: GOSUB 1130: GOTO 660
1390 GOSUB 1400: GOTO 660
```

These changes let you go back to CIRCLE mode automatically after making a menu selection.

Now UNLOCK the old version of TABLET-CODE APPLESOFT that is on your diskette, (the one on which you're putting your own versions) and then SAVE this new version on your diskette. RUN HELLO and start using the GRAPHICS TABLET SOFTWARE. When you want to draw an open circle, press the pen to the square marked SLIDE. Indicate one point for the center of the circle, and another for a point on the perimeter. The circle will be drawn to specification, and you'll remain in CIRCLE mode until you choose another.

Be forewarned that if you make a CIRCLE which is too large for the screen, then you'll get an error. Just press **esc** to get back to DRAW mode. If you don't like this "feature", the following lines will fix the problem:

```
2425 ON ERR GOTO 2480
2470 NEXT TH: ON ERR GOTO 2650
2475 GOTO 2340
2480 PRINT D$;"PR#0": GOSUB 2300: PRINT D$;"PR#";SL: PRINT "M2":
  VTAB 23: HTAB 12: POKE 41, PEEK(41)+4: PRINT "CIRCLE OFF SCREEN.
  RESPECIFY."
2485 GOSUB 1300: PRINT D$;"PR#";SL: PRINT "N,H2": ONERR GOTO 2650
2490 GOTO 2340
```

Note that you can still make circles which go out of the VIEWPORT. There's no easy way to prevent this.

EXAMPLE: DISC MODE

The DISC mode is just the same as CIRCLE mode, except that instead of plotting individual points on the perimeter, you'll have to draw lines across the diameter to fill in the circle. Because they have so much code in common, you can make DISC use much of the code from CIRCLE. Here are the changes to CIRCLE to make it do DISCs, too:

```
2340 CM=5: GOTO 2348
2342 REM ** DISC MODE **
2344 CM=6
2348 GOSUB 1130: PRINT D$;"IN#";SL
2445 IF CM=6 THEN 2464
2462 GOTO 2470
2464 HPLOT X+DX,Y+DY TO X-DX,Y-DY: HPLOT X+DX,Y-DY TO X-DX,Y+DY
2466 HPLOT X+DY,Y+DX TO X-DY,Y-DX: HPLOT X+DY,Y-DX TO X-DY,Y+DX
```

To change the menu vector table so that the SLIDE square will activate CIRCLE and the SEPARATE square will activate DISC, change line 280 to read:

```
280 ON X+1 GOTO 140,300,896,620,1460,550,1150,2160,1380,650,290,1580,
  1680,1740,1840,570,330,420,2342,2330,1970,2070
```

and make this other change:

```
660 ON CM+1 GOTO 170,1580,1680,1740,1840,2330,2342
```

This next line lets you reenter both CIRCLE and DISC modes after you make a menu selection (such as PEN COLOR). Finally, if you added the error handling subroutine described above, then change it so:

```
2390 ON CM-4 GOTO 2330,2342
```

Now again SAVE your modified program under the name TABLET-CODE APPLESOFT on the diskette you're using for your experimentation.

THE FIRMWARE

On the Graphics Tablet Interface card is a 2K byte ROM (Read-Only Memory). This ROM contains all the subroutines which read and interpret the signals from the Graphics Tablet. These subroutines can be used easily from any BASIC program.

The Graphics Tablet Firmware performs many functions. Its main purpose is to read the position of the Tablet's pen on the surface of the Tablet, and return that position in a numerical form to a BASIC program. But, it also does much more:

- It lets you supply horizontal and vertical offset information. It will use this offset information in calculating the pen position. This lets you place the origin (where the X and Y coordinates are both 0) anywhere on the Tablet surface. The offsets can be integers from -32767 to +32767.
- It allows you to give a scaling divisor, from 1 to 32767. You can tell the Tablet firmware to divide all coordinates by this number before it passes them to your BASIC program. This lets you calibrate the Tablet units (200 to the inch) to your own scale.
- It allows you to select among ten different modes on the Apple's screen. Text, low-, and high-resolution graphics (on either Page 1 or Page 2) can be selected, and you can mix text with graphics.
- It automatically displays a flashing cursor on the Apple's screen, given the proper scaling and offset information. Cursors are available or may be suppressed in all screen modes.
- You can tell the Tablet to suppress all output from your Apple.
- You can read not only the position of the pen, but also whether it is within readable distance, whether the pen is up or pressed down, detect pen-down and pen-up movements, and read the keyboard to see if a key has been pressed.

Your programs can communicate to the Firmware subroutines by using the BASIC commands `PR# s` and `IN# s`, where `s` is the number of the peripheral connector slot in the Apple which holds the Tablet Interface card. (The `PR#` command indicates that all subsequent output is to be directed to the Firmware subroutines in a certain slot, and the `IN#` command indicates that all subsequent input is to be taken from the Firmware subroutines in the given slot.) When your program wants to stop talking to the Firmware subroutines, it can issue a `PR#0` or `IN#0` command to direct output or accept input from the normal screen and keyboard.

To avoid alienating DOS (the Disk Operating System), you'll have to issue the `PR#` and `IN#` commands in the form of DOS commands. See the section on Selecting I/O Devices in your DOS manual.

TABLET CONTROL

To send control information to the Tablet, just execute a `PR# s` command from BASIC and PRINT a string of Tablet Control commands. The Control commands will not be displayed on the Apple's screen; they will be used by the Tablet alone.

There are seventeen Tablet Control commands. These commands take the form of a letter or a word, sometimes followed by a number. Commands are executed in a sequential order as given to the tablet by the user.

Commas are used as delimiters between commands and must not begin or end the command string. Spaces are ignored. A null string issued to the tablet is invalid. Only the first alphabetic character of a command is meaningful; the other alphabetic characters are ignored and may be omitted.

Following is a list of Tablet Control commands. The letter "n" that follows some of these commands represents an integer. The Tablet Control commands are:

`TEXT n` Sets the Apple's screen to show text mode. `n` determines which page of Text to display and can be either 1 or 2.

`HGR n` Sets the Apple's screen to show full-screen high-resolution graphics mode. `n` determines which page of graphics to display and can be either 1 or 2.

`LGR n` Sets the Apple's screen to show full-screen low-resolution graphics mode. `n` determines which page of Graphics to display and can be either 1 or 2.

`MLXHGR n` Sets the Apple's screen to show high-resolution graphics mode, mixed with four lines of text at the bottom. `n` determines which page of text and graphics to display and can be either 1 or 2.

`GR n` Sets the Apple's screen to show low-resolution graphics mode, mixed with four lines of text at the bottom. `n` determines which page of text and graphics to display and can be either 1 or 2.

SCALE n

Sets the Tablet scaling divisor to n. All coordinates generated by the Graphics Tablet will be divided by n before they are given to your program. The range for n is 1 to 32767. If you give the Tablet a negative scaling divisor, it will ignore the minus sign and use the positive number. A scale factor of 0 is undefined and will not work.

XOFF n

Sets the Tablet horizontal (X) offset to n. If the R command is enabled, all horizontal coordinates will have n added to them before they are given to your BASIC program. The offset value, n, may range from -32767 to +32767.

YOFF n

Sets the Tablet vertical (Y) offset to n. If the R command is enabled, all vertical coordinates will have n added to them before they are given to your BASIC program (see R command below). The offset value, n, may range from -32767 to +32767.

F

Ignore scaling divisor. None of the coordinates generated by the Tablet will be scaled or offset. The cursor, however, will not ignore scale and offset information.

R

Use scaling divisor. All coordinates generated by the Tablet will be divided by the scaling divisor before they are given to your BASIC program. Then offset values will be added.

AFTER

If the R command is used, the offsets will be added after the scaling operation. This command is turned off (the BEFORE command is reinstated) with any subsequent command which sets a screen mode, including the DEFAULT command.

BEFORE

If the R command is used, the offsets will be added before the scaling operation.

NOPRINT

Disables all on-screen printing. After a NOPRINT command is sent to the Tablet, no new output generated by the Apple will be displayed on the screen. NOPRINT mode is turned off by any other Tablet Control command string or by a BASIC PR#0 command.

CURSOROFF

Turns off the sparkling cursor. The cursor will remain off until any other Tablet Control command is sent which sets a screen mode (the DEFAULT command also turns the cursor on).

P

Sets Stream mode. If the pen is within the proximity of the Tablet, the Tablet Firmware will send coordinates each time it is polled, regardless of the pen position or status. This command is turned off (the Q command



Q



DEFAULT



is reinstated) with any subsequent command which sets a screen mode, including the DEFAULT command.



Resets Stream mode. The Tablet firmware will send coordinates only when it is polled, and the pen is pressed down.



Sets the standard (default) Tablet mode:



- HGR 2 screen mode
- SCALE=16
- XOFF=1536
- YOFF=1536
- F (no scaling or offsets)
- BEFORE
- Q (stream mode off)
- Cursor on
- Printing on



For example, if the Tablet Interface card is in slot number 5 and you want the Tablet to set low-resolution graphics mode, with four lines of Text at the bottom, use a scaling divisor of 16, and use the offsets stored in the variables X0 and Y0, and apply them before the scaling, you would use this Tablet Control command:



PR#5: PRINT "GR 1, SCALE=16, XOFF=""X0""; YOFF=""Y0""; BEFORE":PR#0



Of course, you could shorten it by eliminating extraneous spaces and using only the first letter of each Control command name:



PR#5: PRINT "G1,S16,X":X0";"Y":Y0";"B":PR#0



Since you are using Apple DOS, you must use DOS's PR# command in order to use both DOS and the Tablet. If you've got the slot number of the interface card in the variable SL, then the same Tablet Control command would read:



PRINT D\$;"PR#";SL:PRINT "G1,S16,X":X0";"Y":Y0";"B":PRINT D\$;"PR#0"



It's important not to add a semicolon (;) or comma (,) at the end of the Tablet Control PRINT string. The Tablet will execute the command only when it receives a RETURN character. A semicolon or a comma after the string will suppress the RETURN; therefore, the Tablet will never carry out your commands because it won't hear the end of them.



Any illegal construct in a Control command, including numbers out of range, will cause the screen to return to text mode and the message

*** TABLET SYNTAX ERROR



to appear on the screen.

ACCEPTING INPUT

Once you've told the Tablet what kinds of numbers you expect to be getting from it, you can use the BASIC statements IN# and INPUT to get the pen coordinates and status information from the Tablet.

The Tablet sends its coordinate and status information in this format:

+0000, +0000, +00
X-position Y-position sign status

The X- and Y-position coordinates must be integers from -9999 to +9999. The user is responsible for adjusting the X and Y offsets and the scale value so that values returned by the tablet fall within this range. It is possible to exceed this range, in which case an Applesoft error will be generated. These coordinates indicate the position of the pen on the Tablet. If the R command is in effect, these coordinates indicate the position of the pen plus the offset and divided by the scaler.

The sign and status digits indicate the status of the pen and keyboard. If the sign is negative, then a key has been pressed. The two digits have separate meanings:

00
Tens digit: Ones digit:
0 = pen is on scale 0 = pen is down, and has been down
1 = pen is off-scale 1 = pen was just lifted
 2 = pen was just pressed down

So let's write a program to read the Tablet and print out the coordinates, without scaling or offsets, on the Text screen. Let's assume that the slot number of the Interface card is stored in the variable SL.

```
100 PR#SL: PRINT "T1, F, C, P": PR#0: REM INITIALIZE TABLET
110 IN#SL: INPUT X,Y,Z: IN#0: REM READ TABLET
120 PRINT "THE X-POSITION IS ";X; ", THE Y-POSITION IS ";Y; "."
130 IF Z<0 THEN PRINT "THE KEYBOARD HAS BEEN PRESSED."
140 IF ABS(Z)>=10 THEN PRINT "THE PEN IS OFF-SCALE."
150 Z=ABS(Z): IF Z>=10 THEN Z=Z-10: REM GET ONES DIGIT
160 IF Z=0 THEN PRINT "PEN IS DOWN."
170 IF Z=1 THEN PRINT "PEN WAS JUST LIFTED."
180 IF Z=2 THEN PRINT "PEN WAS JUST PRESSED DOWN."
190 PRINT
200 POKE -16368,0: GOTO 110:REM CLEAR KEYBOARD STROBE, REPEAT
```

This program will work in either Applesoft BASIC or Apple Integer BASIC.

Line 100 sets the Tablet Control parameters. Line 110 gets input from the tablet, and the remaining lines interpret the values and print an explication. Line 200 clears the keyboard strobe (if a key was pressed) and loops back to get another set of values.

This program works in Stream mode, that is, it is continually getting input from the Tablet regardless of the position of the pen. If you change the Tablet Control command string to read

```
100 PR#SL: PRINT "T1, F, C, Q": PR#0
```

then the coordinates will be returned only when the pen is pressed down.

Let's write a subroutine in BASIC which is to return the X and Y coordinates of the next pen press, or return with the variable KY set to 1 if the user presses the **RETURN** key on the keyboard. Let's assume that the Tablet has been initialized in the main program (see previous example, line 100).

```
200 REM ** SUBROUTINE TO GET A PEN PRESS OR KEYPRESS **
210 KY=0: REM FLAG FOR KEYPRESS
220 IN#SL: INPUT X,Y,Z: IN#0
230 IF Z=2 THEN RETURN: REM PEN DOWN
240 IF Z>0 THEN 280: REM NO KEYPRESS
250 K=PEEK(-16384): REM GET KEYPRESS
260 IF K<>13 THEN 280: REM IS IT A RETURN?
270 KY=1: RETURN: REM YES, IT IS.
280 POKE -16368, 0: GOTO 220: REM NO, KEEP LOOKING.
```

FROM MACHINE LANGUAGE

You can perform the same Tablet operations from within a machine language program that you can from a BASIC program. Even though machine language programs are a little more difficult to write, they will run faster and use less memory than their BASIC counterparts.

Your machine language programs will invoke the various functions of the Graphics Tablet firmware by performing JSR (Jump to SubRoutine) operations to subroutines inside the Tablet's ROM, rather than using the PR# and IN# statements in BASIC. Your programs will pass information to the Tablet by storing it in fixed locations in memory, and will receive information from the Tablet by storing it in other fixed locations, instead of using PRINT and INPUT statements as a BASIC program would.

Since the Tablet firmware operates in the same manner regardless of whether it is being driven by a BASIC or a machine language program, this section will explain only the specifics of machine language operation of the Tablet. For a description of the modes and parameters which the Tablet firmware recognizes, please see the previous section.

The Tablet firmware is absolutely located in the Apple's memory at locations \$C800 through \$CFFF. This is a 2K memory space which is shared by all peripherals, and can be used by any one peripheral card at any time. In order to let the Graphics Tablet card take possession of this common ROM space, you must reference two special memory locations. First, you must reference location \$CFFF. This will turn off all interface cards which may be using the common ROM space. Then you must make at least one reference to any address in the range \$Cn00 through \$CnFF, where n is the number (from 0 to 7) of the peripheral connector slot which holds the Graphics Tablet interface card. Once this is done, the Tablet's ROM will be placed into its proper memory range and you can reference its subroutines normally.

After you activate the ROM, you should store the slot number of the Graphics Tablet (in the format \$Cn) in location \$07F8. This lets other Apple programs know that the Tablet is active and in use.

Subroutine POINT (location \$Cn02) lets you read a single point from the Tablet. The coordinates of the point, along with the pen status information, will be stored as a 15-character long ASCII string, beginning at location \$0200 and ending with a RETURN code at location \$020E. The format of this string is described in the previous section called ACCEPTING INPUT.

The subroutine DEFAULT (location \$C9E0) sets all the Graphics Tablet parameters and modes to their default values. It operates the same as the Tablet control command DEFAULT.

The subroutine MREAD (location \$CBB9) allows you to read the pen position and status quickly, and get the result in binary (rather than ASCII, as POINT does). It returns the X and Y coordinates in the following locations:

XFFL	\$0281	Lower byte of X-coordinate
XFFH	\$0282	Upper byte of X-coordinate
YFFL	\$0283	Lower byte of Y-coordinate
YFFH	\$0284	Upper byte of Y-coordinate
TEM	\$0280	Pen status

The X and Y coordinates are numbers from -32767 to +32767. Notice that this is a greater range than the coordinates passed by POINT. The numbers are in two's compliment form, and the high bit of the upper byte of each coordinate determines the sign of that coordinate. The pen status byte is interpreted much the same as it is for POINT: the lower 4 bits represent the pen status and the upper bit represents the keyboard status.

The SCALE subroutine (location \$CB70) is normally called immediately after MREAD. It performs a scaling and offset operation on the X and Y coordinates generated by MREAD and places the results in these four locations:

TEMXL	\$0285	Lower byte of scaled X-coordinate
TEMXH	\$0286	Upper byte of scaled X-coordinate
TEMYL	\$0287	Lower byte of scaled Y-coordinate
TEMYH	\$0288	Upper byte of scaled Y-coordinate

These values are also in two's compliment form and range from -32767 to +32767.

The CURSOROUT subroutine (location \$C8F0) is normally called immediately after an MREAD. CURSOROUT calls SCALE and uses the scaled results to place a cursor on the Apple's screen. The cursor is placed by an exclusive-OR operation, so another call to CURSOROUT using the same coordinates will remove the cursor and leave the screen unchanged.



The CURSOROUT subroutine places the cursor on the screen which the Tablet was told to display. It is not necessarily the screen which the Apple is currently displaying. If you manually change the screen setting after calling DEFAULT or setting the Tablet PAGE parameter (see below), then the Apple may be displaying a video mode which is different from the one in which the Tablet is displaying a cursor.

You can pass parameters to the Tablet firmware by storing the proper values in special memory locations. Here are the locations used by the Tablet firmware.

The MSLOT parameter (location \$07F8) contains the number of the slot (in the format \$Cn) into which the Graphics Tablet Interface card is plugged.

The PAGE parameter (location \$03B8+MSLOT) holds the code for the current video mode:

\$20	high-resolution page 1	\$40	high-resolution page 2
\$01	low-resolution page 1	\$02	low-resolution page 2
\$21	Mixed high-resolution page 1	\$42	Mixed high-resolution page 2
\$05	Mixed low-resolution page 1	\$0A	Mixed low-resolution page 2
\$04	Text page 1	\$08	Text page 2
\$00	No cursor		

If you set the high bit of the PAGE byte, then the scale and offset factors will be applied.

The MPAGE parameter (location \$0438+MSLOT) holds some of the same information as the PAGE parameter. The lower six bits of MPAGE are derived from the lower six bits of PAGE exclusive-ORed with the constant \$25. The upper two bits represent the A, B, P, and Q parameters:

Bit 7 ON: Stream mode on Bit 7 OFF: Stream mode off
Bit 6 ON: Offset after scale Bit 6 OFF: Offset before scale

The scale and offset parameters are stored in the following locations:

SCALL	\$0488+MSLOT	Lower byte of scaling divisor
SCALH	\$0538+MSLOT	Upper byte of scaling divisor
OFFXL	\$0588+MSLOT	Lower byte of X-offset
OFFXH	\$0638+MSLOT	Upper byte of X-offset
OFFYL	\$0688+MSLOT	Lower byte of Y-offset
OFFYH	\$0738+MSLOT	Upper byte of Y-offset

The scaling divisor is a binary integer from 0 to 32767. The offsets are two's compliment binary numbers from -32767 to +32767.

QUICK DRAW

The QUICK-DRAW program is a machine language subroutine which acts as an intermediary between the Tablet Firmware and an Applesoft program. Since an Applesoft program using HPLOT cannot draw on the High-Resolution screen fast enough to keep up with the movements of the pen across the Tablet, the QUICK-DRAW subroutine talks directly to the Tablet and plots the points on the high-resolution screen. QUICK-DRAW also makes the points plotted available to the Applesoft program.

QUICK-DRAW must run on an Apple with at least 16K bytes of memory, the Applesoft II BASIC programming language in ROM or the Language System, and a Graphics Tablet Interface card. The Graphics Tablet Firmware must be activated by an IN# command before QUICK-DRAW can be called.

The QUICK-DRAW subroutines are hidden inside an Applesoft program. When you RUN QUICK-DRAW, the Applesoft program will store the subroutines in the memory range \$C00-\$FFF (decimal 3072-4095). The entry point for the subroutines will be placed in memory locations \$2F0 and \$2F1 (decimal 752 and 753). Your Applesoft program, which you will RUN right after you RUN QUICK-DRAW, can PEEK at these locations and get the entry point by executing this line:

```
100 EP% = PEEK(752)+256*PEEK(753)
```

The QUICK-DRAW subroutine deals directly with four Applesoft variables. When you CALL the QUICK-DRAW subroutines, it takes the coordinates of the points it receives from the Tablet and places them in the two Applesoft arrays X% and Y%. It uses the Applesoft variable



N% as an index into these arrays. The subroutine also uses the contents of the variable D% as a DELTA value. It is the QUICK-DRAW subroutine which controls the DELTA and Audio Feedback features of the Tablet software.



You must dimension the arrays X% and Y% prior to calling QUICK-DRAW. Also, you must assign a non-zero value to D%. The D% value is used as described in the DELTA function in Chapter 2; if the value of D% is negative, then the Audio Feedback feature will be turned off.



The QUICK-DRAW subroutine will return control to the Applesoft program under any of four conditions:



- 1) A key on the keyboard was pressed before the pen was pressed down.
- 2) The pen was moved to a place on the Tablet which does not correspond to a position in the current VIEWPORT.
- 3) The pen was lifted after being pressed down inside the VIEWPORT.
- 4) There is no more room in the arrays X% and Y% to store coordinate values.



When one of these conditions arises, the code for that termination condition will be stored in location \$2BC (decimal 700) and control will be returned to the Applesoft program.



You can define a VIEWPORT for the QUICK-DRAW subroutines by storing:



- the coordinate of the left edge in locations 3089 and 3090;
- the coordinate of the right edge plus one in locations 3091 and 3092;
- the coordinate of the top edge in location 3093; and
- the coordinate of the bottom edge plus one in location 3094.



See lines 1100 and 1120 of the TABLET-CODE APPLESOFT program for an example of how to pass VIEWPORT coordinates to the QUICK-DRAW subroutine.



BY ANY OTHER NAME



You can change the names of the variables which QUICK-DRAW will use by executing a special CALL to QUICK-DRAW. Normally, QUICK-DRAW uses these variable names:



D% for the DELTA value
X% for the X-coordinate array



N% for the index to the arrays
Y% for the Y-coordinate array

You can change these four variable names to be whatever you like. However, they must always be of the integer variable type (denoted by the percent sign (%) following the name). To rename the variables, use this format:

220 CALL EP%, DELTA%, NUMBER%, XVAL%, YVAL%

Since Applesoft only recognizes the first two letters of a variable name, this will make QUICK-DRAW use the variable DEX for its DELTA, NUX for N%, XV% for X%, and YV% for Y%. If you want to change only one of the names, just leave the others out, but keep the proper number of commas:

230 CALL EP%, IN%,

will make QUICK-DRAW use the variable IN% instead of N%. You must keep the variable names in the order DX%, N%, X%, Y%.



APPENDIX A USE AND CARE

68	Care of the Menu Overlay
68	Care of the Tablet
69	Care of the Interface
69	If It Doesn't Work

CARE OF THE MENU OVERLAY

You can write on the clear plastic menu overlay with most anything: soft (Number 2) pencils, felt-tip pens, permanent markers, crayons, and the like. However, ball-point pens tend not to write well on the overlay, and colored or hard lead pencils also have problems.

You can wipe the overlay clean of most marks or doodles you have drawn using a soft cloth and a mild soap-and-water solution. Most marks from felt-tip or "permanent" markers can be removed easily. Some markers, however, will leave truly permanent scars on the overlay; it's a good idea to test any marker on an inconspicuous corner of the overlay before you use it to draw all over your Tablet. To be safe, use felt-tip markers designed for use on acetate or mylar (or for use with overhead projectors). These will give you visible, non-smearing colors, but the marks will wipe off without a trace.

If you are getting inexplicable "glitches" on your screen you probably have a static problem. The solution is simple: Wipe the overlay with the static cloth that came with your Graphics Tablet. A treatment with the cloth should remove any excess static from the overlay.

CARE OF THE TABLET

Your Graphics Tablet is constructed of a solid wood base, protected below by a sheet metal baseplate and above by a molded, snap-on plastic cover. If the top cover gets dirty, it can be cleaned with a soft cloth and a mild soap-and-water solution. Don't use any abrasives or strong detergents on the surface or case of the Tablet: they may scratch or damage the plastic. If possible, keep the Tablet covered when you aren't using it.

DON'T leave anything which has a strong magnetic field on or near the Tablet. This will disrupt its natural magnetic orientation and make it malfunction. Keep your diskettes off the Tablet! Its magnetic field may alter or erase the information on them. Don't place disk drives, televisions, electric motors, magnets, or large, heavy metallic objects on top of the Tablet.

Keep the Tablet in a cool, dry place. Don't leave it in direct sunlight or in a car trunk or some other hot, stuffy place. Too much heat will warp its cover.

Be careful with the Tablet when you're moving it from place to place. Don't drop it or jar it. Even though it's pretty solid, it can be seriously damaged by a bad fall.



CARE OF THE INTERFACE

The Interface card is really the most delicate part of the Graphics Tablet. When inserting, removing, or transporting it, be very careful not to bend any of its pins or components. To be safe, always carry it in the box in which it was shipped, nestled in protective foam. Keep it away from strong electrical or magnetic fields, and don't even think of touching it if there's a lot of static electricity in the area.

If you've been inserting and removing the Interface card into the Apple a lot, then it's possible that the metal "fingers" have gotten dirty and are not making good contact with the Apple. In this case, the easiest way to clean the fingers is to just use an ordinary pencil eraser and rub all of the gunk off. If you want to be thorough, use cotton swabs and rubbing alcohol to clean the fingers on the card.



IF IT DOESN'T WORK

If you've exposed your Tablet to bad magnetic influences or it's been bumped and jarred a lot, it may develop "dead spots" on its surface, spots where the pen won't draw. These aren't permanent, they're just a loss of magnetic orientation in certain spots of the Tablet. Take the Tablet to your Apple service center. The service center should have the proper equipment to reorient your errant Tablet and make it work again.





APPENDIX B BACKING UP YOUR DISKETTES

72 With Two Disk Drives
72 With One Disk Drive



WITH TWO DISK DRIVES

If your Apple has two disk drives, you can easily make a copy of either GRAPHICS TABLET SOFTWARE diskette by using the diskette copying program on your Apple SYSTEM MASTER diskette. You will need three diskettes:

- 1) One of the GRAPHICS TABLET SOFTWARE diskettes, enclosed with your Tablet;
- 2) The SYSTEM MASTER diskette, enclosed with your Disk II; and
- 3) A blank, uninitialized diskette. If you like, you can use a preinitialized diskette, but all information on that diskette will be destroyed.

Boot your system using the SYSTEM MASTER diskette (see your DOS manual, or, if you have an Autostart ROM, see your Autostart ROM manual) and type

RUN COPY

After the disk drive stops whirring, place the GRAPHICS TABLET SOFTWARE backup diskette in one drive, and place the blank diskette in the other. The GRAPHICS TABLET SOFTWARE diskette will be the "Original", and the blank diskette will be the "Duplicate". Follow the instructions in the section on using the COPY program in your DOS manual.

Once you've copied the diskette, label the duplicate so you'll know what it is. Then put the original away in a safe place. If you ever lose or destroy the duplicate, then before you start to use the original, make another copy of it. It's also a good idea to periodically make duplicate copies of the diskettes which hold your pictures.

WITH ONE DISK DRIVE

If your Apple has only one disk drive, then you'll have to copy all the programs which comprise the GRAPHICS TABLET SOFTWARE package one by one, loading each program from the original diskette and saving it to the duplicate. It's a lengthy procedure, but well worth your trouble.

Boot your system using the GRAPHICS TABLET SOFTWARE diskette. Press **ESC** to get to the HELLO menu, select **0** to QUIT, and press **RETURN**. Now remove the diskette and write-protect it by sticking a write-protect tab (a thin but sturdy strip of tape will do) over the

square notch on the left side of the diskette. This is important! It will prevent you from accidentally destroying anything on the original diskette. Now insert a blank, uninitialized diskette in the drive. You can use a preinitialized diskette, but all information on it will be destroyed. Type

INIT HELLO

and press **RETURN**. You're now initializing the diskette with the HELLO program from the GRAPHICS TABLET SOFTWARE diskette. This takes about a minute.

Now switch to the GRAPHICS TABLET SOFTWARE diskette and type **LOAD MENU ALIGNMENT**

Now switch to the duplicate diskette and type **SAVE MENU ALIGNMENT**

Now switch to the GRAPHICS TABLET SOFTWARE diskette and type **LOAD TABLET-CODE APPLESOFT**

Now switch to the duplicate diskette and type **SAVE TABLET-CODE APPLESOFT**

Now switch to the GRAPHICS TABLET SOFTWARE diskette and type **LOAD QUICK-DRAW**

Now switch to the duplicate diskette and type **SAVE QUICK-DRAW**

Now switch to the GRAPHICS TABLET SOFTWARE diskette and type **BLOAD UTILITIES**
BLOAD GRAPHICS TABLET LOGO

Now switch to the duplicate diskette and type **BSAVE UTILITIES, A\$6000, L\$330**
BSAVE GRAPHICS TABLET LOGO, A\$2000, L\$2000

Now enter this program:

NEW

```
10 D$=CHR$(4)
20 PRINT D$;"OPEN GRAPHICS TABLET SOFTWARE"
30 PRINT D$;"WRITE GRAPHICS TABLET SOFTWARE"
40 PRINT "RUN QUICK DRAW"
50 PRINT "RUN TABLET-CODE APPLESOFT"
60 PRINT D$;"CLOSE GRAPHICS TABLET SOFTWARE"
70 END
```

RUN

This short program creates an EXEC file called GRAPHICS TABLET SOFTWARE, whose function is to set up the Apple to RUN the programs which make the Graphics Tablet work. You need to have this file on every duplicate diskette you make; if you're going to be making many duplicate copies, you might want to SAVE this short program so you don't have to retype it every time you need it.

To SAVE this program, type

SAVE FILEMAKER

Then, whenever you're making a duplicate, put in this diskette and type

LOAD FILEMAKER

put in the duplicate diskette, and type

RUN

That's all there is. Once you've copied the diskette, label the duplicate so you'll know what it is and put it away in a safe place. If you ever lose or destroy the original, then before you start to use the duplicate, make another copy of it.

APPENDIX C COLOR ANOMALIES

76 Unusual Color Effects...

77 ...And How to Get Them



UNUSUAL COLOR EFFECTS...

You may have already noticed that a few strange things happen when you try to use certain combinations of colors with the Graphics Tablet. Don't worry: these are normal, predictable phenomena which are caused not by the Tablet, but by the Apple itself.

The Graphics Tablet displays its pictures using the Apple's high-resolution graphics mode. In this mode, there are 53,760 individual dots on the screen, and six colors (black, white, orange, blue, green, and violet). The Apple should therefore need several hundred thousand individual "bits" of information to form a picture. But the Apple uses only 65,536 bits of information (organized into 8,192 eight-bit "bytes") to form the picture! The reason the Apple can display such complex pictures using so little memory is the same reason that sometimes the colors don't appear normal: not all colors can be used in all places on the screen, and each dot is limited in the number of colors it can be.

This specialization of function causes some combinations of colors to work differently than you might expect. There are three different effects which are caused by the limitation in the color scheme. They occur in all drawing modes, but only where one color borders another and the borderline is not horizontal. For example, the color problems could occur on the two vertical sides of a FRAME but not on the top or bottom. Here are the three effects:

- 1) DASHED LINES. When you draw black or white lines on a colored field (or vice versa), non-horizontal lines will tend to become dashed and incomplete, and vertical lines may not appear at all.
- 2) ZEBRA STRIPES. When you draw colored lines on a colored field, non-horizontal lines don't appear their normal colors, but instead are sometimes black-and-white striped. Vertical lines will appear either completely black or completely white.
- 3) COLOR FLIP. When you draw with one color (or black or white) across a colored field, sometimes a seven-dot wide area around a non-horizontal line will change color. This will result in a colored "shadow" appearing around the line.

These effects occur in various combinations, depending upon the colors you use.



...AND HOW TO GET THEM

The table on the next page illustrates seven different combinations of the effects mentioned above, and what color combinations produce which effects. To use the table, find the pen color you're using along the left side of the table. Then look on the top edge of the table and find the color of the area on the screen across which you want to draw. Where the row for the pen color and the column for the field color intersect, there's a number. Find the number in the legend to the table and read about the effect you'll get.

About BLACK1, WHITE1, BLACK2, and WHITE2: Due to the vagaries of the Apple's color generation scheme, there are two instances each of the colors black and white. When you look at the Tablet color menu (see the section on PEN COLOR in Chapter 2), you'll see that there are two black squares and two white squares along with the four colored squares. The black and white in the top row are BLACK1 and WHITE1; the ones in the bottom row are BLACK2 and WHITE2. The reason for the duplication is that the 1's cause fewer problems when used with green and violet than do the 2's, and similarly the 2's go better with blue and orange than do the 1's. When this book refers to black or white, it means BLACK1 or WHITE1.

PEN COLOR

	BLACK1	GREEN	VIOLET	WHITE1	BLACK2	ORANGE	BLUE	WHITE2
BLACK1	0	2	2	1	0	2	2	1
GREEN	2	0	3	2	4	5	6	4
VIOLET	2	3	0	2	4	6	5	4
WHITE1	1	2	2	0	1	2	2	0
BLACK2	0	2	2	1	0	2	2	1
ORANGE	4	5	6	4	0	0	3	2
BLUE	4	6	5	4	2	3	0	2
WHITE2	1	2	2	0	1	2	2	0

FIELD COLOR

Color Effects Table

LEGEND:

- 0: No effect.
- 1: Colors appear as expected; no anomalies.
- 2: DASHED LINES on non-horizontal lines; vertical lines may disappear.
- 3: ZEBRA STRIPING on non-horizontal lines; vertical lines appear solid black or white.
- 4: DASHED LINES with a COLOR FLIP.
- 5: Pure COLOR FLIP: non-horizontal lines appear "chunky" and wider than normal.
- 6: ZEBRA STRIPING with a COLOR FLIP.

APPENDIX D

PROGRAM LISTINGS

80	Tablet-Code Applesoft
85	Variable Atlas
87	Subroutines
88	Special Locations
89	ROM Code
108	Quick-Draw
118	Utilities

TABLET-CODE APPLESOFT

```

10 REM * TABLET SOFTWARE. COPYRIGHT APPLE 1979. B. EHLERS *
20 LOMEM 25392
30 DS = CHR$(4): PRINT DS: "CLOSE GRAPHICS TABLET SOFTWARE"
40 ONERR GOTO 2610
50 DS = CHR$(4): PRINT DS: "OPEN TAB INFORMATION.D1": PRINT DS: "READ TAB
INFORMATION": INPUT SL: INPUT XL: INPUT YL: INPUT XH: INPUT YH
60 PRINT : PRINT DS: "CLOSE TAB INFORMATION"
70 ONERR GOTO 2650
80 EP% = PEEK(752) + 256 + PEEK(752) MX = 800
90 DIM YM(MX), XM(MX)
100 PRINT DS: "BLLOAD UTILITIES/A$4000,D1"
110 XA = XH - XL: YA = YH - YL: LT = INT((XA + YA) / 2): PI = INT(LT / 11)
120 SO = INT(XA / 11 + 5)
130 MD = INT(PI / 2): XM = XL: YM = 2 + MD + YL
140 HGR2 PC = 3: DC = 0: HCOLOR= PC: W = 1: DF = 1
150 X1 = XM * 2: Y1 = YM * 2: X2 = (INT((XH + 2 - X1) / 280 + 5) * 280) Y
2 = INT(X2 * 192 / 280): X5 = X1 - X6 = X2: Y5 = Y1 - Y6 = Y2
160 DX = -2: SM = S2: GOSUB 1070: RD = 0
170 CM = 0: N% = 1: CALL EP%: CD = PEEK(700): ON CD + 1 GOTO 190, 200, 170, 1
180 GOTO 170
190 PRINT : PRINT DS: "PR#0": PRINT DS: "IN#0": GET A$: IF ASC(A$) < 0 OR 2
7 THEN 200
192 TEXT : HOME : VTAB 12: HTAB 13: INPUT "QUIT? (Y OR N)": A$: IF A$ = "Y"
THEN HOME : VTAB 12: HTAB 10: PRINT "LOADING HELLO PROGRAM": POKE
104, 8: POKE 103, 1: PRINT DS: "RUN HELLO.D1": STOP
194 GOSUB 1130: GOTO 170
200 IF PEEK(640) < 0 THEN 170
210 POKE 640, 0
220 XF = XL * 2: YF = YL * 2: SF = SO: GOSUB 2590: REM SENSE
MENU
230 PRINT : PRINT DS: "IN#": SL
240 INPUT X, Y, Z: IF Y > 1 THEN GOSUB 1130: GOTO 170
250 IF Y < 1 AND Y > = 0 THEN PRINT DS: "PR#0": PRINT CHR$(7): PRINT
: PRINT DS: "PR#": SL: PRINT "N": ON Y + 1 GOTO 280, 290
260 GOTO 200
270 TEXT : PR#0: PRINT "ERROR": STOP
280 ON X + 1 GOTO 140, 300, 690, 620, 1460, 550, 1150, 2150, 1380, 650, 290, 1580, 16
80, 1740, 1340, 570, 330, 420, 2490, 2330, 1970, 2070
290 GOSUB 1130: GOTO 170
300 IF XT = X3 AND YT = Y3 AND X4 = XB AND Y4 = YB THEN HCOLOR= 3C: HPLOT
0, 0: CALL 62454: GOTO 520
310 HCOLOR= 3C: HPLOT X3, Y3: FOR T1 = Y3 TO Y4: HPLOT X3, T1 TO X4, T1: NEXT
HCOLOR= PC: GOSUB 1130: GOTO 170
320 TEXT : PRINT DS: "PR#0": HOME : VTAB 7: HTAB 6: PRINT "PLEASE TYPE THE
PICTURE NAME": PRINT : HTAB 7: PRINT DS: "IN#0": INPUT "=>": A$: IF
A$ = "" THEN GOSUB 1130: GOTO 170
330 VTAB 9: HTAB 37: CALL = 868: HTAB 1: GOSUB 530
345 B$ = "PIC": B$ = ONERR: GOTO 400
350 PRINT DS: "BLLOAD ", B$, ", A$4000, V0, D", CS
360 S2 = PEEK(16632) + 256 + PEEK(16633): IF S2 < 0 THEN 130
370 X1 = PEEK(16504) + 256 + PEEK(16505): X2 = PEEK(16506) + 256 + PEEK
(16507): Y1 = PEEK(16508) + 256 + PEEK(16509): Y2 = PEEK(16510) +
256 + PEEK(16511)
380 ONERR GOTO 2650
390 HOME GOTO 520
400 B$ = RIGHTS(B$, (LEN(B$) - 4)): ONERR GOTO 2650
410 PRINT DS: "BLLOAD ", B$, ", A$4000, V0, D", CS: GOTO 150
420 TEXT : PRINT DS: "PR#0": HOME : VTAB 7: HTAB 3: PRINT "PLEASE TYPE A N
AME FOR THIS PICTURE": PRINT : HTAB 7: PRINT DS: "IN#0": INPUT "=>": A$:
IF A$ = "" THEN GOSUB 1130: GOTO 170
430 VTAB 9: HTAB 37: CALL = 868: HTAB 1: GOSUB 530
440 GOSUB 1330: HCOLOR= BC: GOSUB 1040: H = INT(X1 / 256): POKE 16504, H
POKE 16505, X1 - H * 256: H = INT(X2 / 256): POKE 16506, H: POKE 1650
7, X2 - H * 256: H = INT(Y1 / 256): POKE 16508, H: POKE 16509, Y1 - H +
256
450 H = INT(Y2 / 256): POKE 16510, H: POKE 16511, Y2 - H * 256: H = INT(S
2 / 256): POKE 16632, H: POKE 16633, S2 - H * 256

```

```

460 B$ = "PIC": B$ = ONERR: GOTO 490
470 HCOLOR= PC: PRINT DS: "VERIFY": B$, ", D", CS: ONERR GOTO 2630
480 VTAB 21: HTAB 1: PRINT "A PICTURE ALREADY EXISTS WITH THAT NAME": PRINT
HTAB 12: INPUT "CONTINUE (Y OR N)": E$: IF E$ < 0 OR "Y" THEN 510
490 ONERR GOTO 2650
500 PRINT DS: "BSAVE": B$, ", A$4000, L$1FFB, V0, D", CS
510 HOME : PRINT DS: "PR#": SL: PRINT "H2, N"
520 GOSUB 1070: PRINT DS: "IN#": SL: GOTO 170
530 VTAB 10: CALL = 958: PRINT : HTAB 16: PRINT "DRIVE # ? (DEFAULT)""
, DF": HTAB 25: INPUT "": CS: IF CS < 0 OR "1" AND CS < 0 OR "2" THEN
CS = STR$(DF)
540 DF = VAL(C$): VTAB 11: HTAB 24: CALL = 958: PRINT CS: RETURN
550 REM *** SOFT RESET COMMAND ***
560 GOSUB 1330: GOSUB 1070: DX = -2: GOTO 170
570 TEXT : PRINT DS: "PR#0": HOME : PRINT DS: "IN#0": GOSUB 530
580 HOME : HTAB 7: PRINT "PRESS SPACE BAR TO CONTINUE": POKE 34, 2
590 PRINT DS: "CATALOG D", CS
600 POKE -16368, 0: GET A$: IF A$ < 0 OR " " THEN 600
610 GOSUB 1130: GOTO 170
620 REM *** BACKGROUND AND PEN COLOR ***
630 T1 = PC: GOSUB 670: IF PC = 8 THEN PC = T1: GOSUB 1130: GOTO 660
640 BC = PC: PC = T1: HCOLOR= BC: HPLOT 0, 0: CALL 62454: GOTO 520
650 T3 = PC: GOSUB 670: GOSUB 1130: IF PC = 8 THEN PC = T3
660 ON CM + 1 GOTO 170, 1580, 1680, 1740, 1840
670 XF = XM * 2: YF = YM * 2: SF = INT((XH - XM) / 140)
680 PRINT DS: "PR#0": TEXT : HOME : PRINT DS: "PR#": SL: PRINT "G1, R, X", XF, "
, Y", YF": SF: GR : HOME : VTAB 22: HTAB 9: PRINT "CONSTRUCTING COLOR
R MENU"
690 COLOR= 5: FOR Z2 = 0 TO 39: HLIN 0, 39 AT Z2: NEXT
700 XB = 9: YB = 17: X9 = 2: Y9 = 2: C9 = 0: GOSUB 880: X9 = 11: Y9 = 2: C9 = 12:
GOSUB 880: X9 = 20: Y9 = 2: C9 = 3: GOSUB 880: X9 = 29: Y9 = 2: C9 = 15: GOSUB
880
710 X9 = 2: Y9 = 21: C9 = 0: GOSUB 880: X9 = 11: Y9 = 21: C9 = 9: GOSUB 880: X9 =
20: Y9 = 21: C9 = 5: GOSUB 880: X9 = 29: Y9 = 21: C9 = 15: GOSUB 880
720 HOME : VTAB 22: HTAB 7: PRINT "USE THE PEN TO PICK A COLOR": PRINT
: PRINT DS: "PR#": SL: PRINT "N"
730 PRINT DS: "IN#": SL: INPUT X, Y, Z: IF Z < 0 THEN PRINT DS: "IN#0": GET A
$: X = 0: Y = 0: PRINT : IF ASC(A$) = 13 THEN PC = 8: RETURN
740 IF Z < 0 OR 2 THEN 730
750 X = INT(X / 7): Y = INT(Y / 4)
760 IF Y < 2 OR Y > 37 OR Y = 19 OR Y = 20 OR X < 2 OR X > 37 THEN 730
770 PRINT DS: "PR#0": PRINT CHR$(7): IF Y < 1 AND Y > 19 THEN ON INT(X
- 2) / 9 + 1 GOTO 790, 800, 810, 820
780 ON INT((X - 21) / 9) + 1 GOTO 830, 840, 850, 860
790 PC = 0: B$ = "BLACK1": GOTO 870
800 PC = 1: B$ = "GREEN": GOTO 870
810 PC = 2: B$ = "VIOLET": GOTO 870
820 PC = 3: B$ = "WHITE1": GOTO 870
830 PC = 4: B$ = "BLACK2": GOTO 870
840 PC = 5: B$ = "ORANGE": GOTO 870
850 PC = 6: B$ = "BLUE": GOTO 870
860 PC = 7: B$ = "WHITE2"
870 HOME : VTAB 22: HTAB (40 - LEN(B$)) / 2: PRINT B$: FOR Z2 = 1 TO 50
0: NEXT: HCOLOR= PC: HOME : RETURN
880 COLOR= C9: FOR Z2 = 1 TO XB: VLIN YB, Y9 + YB - 1 AT X9: X9 = X9 + 1: NEXT
: RETURN: REM COLOR BOX LO-RES DRAW
890 REM *** WINDOW COMMAND ***
900 PRINT : PRINT DS: "PR#": SL: PRINT "T1, F, C": PRINT DS: "PR#0"
910 TEXT : HOME : VTAB 9: HTAB 15: PRINT "PRESS PEN AT": PRINT : HTAB B: INVERSE
920 PRINT DS: "PR#": SL: PRINT "N, C": PRINT DS: "IN#": SL: INPUT X, Y, Z: IF Z <
0 THEN PRINT DS: "IN#0": GET A$: IF ASC(A$) = 68 THEN GOSUB 1330: HCOLOR=
930 IF Z < 0 THEN IF ASC(A$) = 13 THEN GOSUB 1130: GOTO 170
940 IF Z < 0 OR 2 THEN 920
950 PRINT DS: "PR#0": IF X < XM + 2 OR Y < YM + 2 THEN VTAB 22: HTAB 4: PRINT
CHR$(7): "PLEASE STAY WITHIN THE WORK-AREA": FOR Z2 = 1 TO 500: NEXT
Z2: VTAB 22: CALL = 868: GOTO 920
960 VTAB 11: CALL = 868: HTAB 9: PRINT "UPPER-LEFT AND": INVERSE : PRINT
"LOWER-RIGHT": NORMAL
970 PRINT DS: "PR#": SL: PRINT "N, C": INPUT TX, TY, Z: IF Z < 0 THEN PRINT D
$: "IN#0": GET A$: IF ASC(A$) = 68 THEN GOSUB 1330: HCOLOR= BC: GOSUB
1040: X1 = X5: X2 = X6: Y1 = Y5: Y2 = Y6: GOSUB 1070: GOTO 170
980 IF Z < 0 THEN IF ASC(A$) = 13 THEN GOSUB 1130: GOTO 170
990 IF Z < 0 OR 2 THEN 970

```

```

1000 PRINT : PRINT DS;"PR#0": IF TX > XH * 2 OR TY > YH * 2 THEN VTAB 22
    HTAB 4: PRINT CHR$ (7): "PLEASE STAY WITHIN THE WORK AREA." FOR ZZ
    = 1 TO 500: NEXT ZZ: VTAB 22: CALL - 868: GOTO 970
1010 IF TX < X OR TY < Y THEN VTAB 22: HTAB 5: PRINT CHR$ (7): "PLEASE S
    pecify points correctly!": FOR ZZ = 1 TO 500: NEXT ZZ: GOTO 910
1020 X1 = X: X2 = TX - X1 + 1: Y1 = Y: Y2 = TY - Y1 + 1
1030 GOSUB 1330: HCOLOR= BC: GOSUB 1040: GOSUB 1070: GOTO 170
1040 XT = INT (GF): XB = 279 - XT: YT = INT (HF): YB = 191 - YT: IF XT > =
    2 THEN HPLOT XT - 1, YT TO XT - 1, YB: HPLOT XT - 2, YT TO XT - 2, YB: HPLOT
    XB + 1, YT TO XB + 1, YB: HPLOT XB + 2, YT TO XB + 2, YB
1050 IF YT > = 1 THEN HPLOT XT, YT - 1 TO XB, YT - 1: HPLOT XT, YB + 1 TO
    XB, YB + 1
1060 RETURN
1070 T1 = X2 / 280: T2 = Y2 / 192: IF T1 < T2 THEN S2 = T2: IF INT (T2) <
    T2 THEN S2 = INT (T2) + 1
1080 IF T1 > = T2 THEN S2 = T1: IF INT (T1) < T1 THEN S2 = INT (T1) +
    1
1090 RD = 0: GOSUB 1130: HCOLOR= 0: IF BC = 0 OR BC = 4 THEN HCOLOR= 3
1100 GOSUB 1040: HCOLOR= PC: B1 = INT (XT / 256): B2 = XT - B1 * 256: B3 =
    INT ((XB + 1) / 256): B4 = (XB + 1) - B3 * 256: B5 = YT - B6 = YB + 1: XB
    = INT (XT): X4 = INT (XB): Y3 = INT (YT): Y4 = INT (YB)
1110 GOSUB 1330: HCOLOR= PC: WM = 1: WS = "": ONERR: GOTO 2630
1120 BX = 3069: POKE BX, B2: POKE BX + 1, B1: POKE BX + 2, B4: POKE BX + 3, B3
    : POKE BX + 4, B5: POKE BX + 5, B6: RETURN
1130 IF RD > 0 THEN GOSUB 1400: RETURN
1140 GF = (280 - X2 / S2) / 2: HF = (192 - Y2 / S2) / 2: XF = INT (X1 - GF +
    S2): YF = INT (Y1 - HF * S2): SF = S2: GOSUB 2590: RETURN
1150 REM *** VIEWPORT COMMAND ***
1160 HCOLOR= 3: IF BC = 3 OR BC = 7 THEN HCOLOR= 0
1170 GOSUB 1330: XF = INT (X1 - GF * S2): YF = INT (Y1 - HF * S2): SF = S2
    : GOSUB 2390
1180 GOSUB 1290: GOSUB 1310
1190 IF Z < 0 THEN PRINT DS;"IN#0": PRINT : GET AS: IF ASC (AS) = 13 THEN
    GOSUB 1130: GOSUB 1330: HCOLOR= PC: GOTO 170
1200 IF Z < 0 THEN IF ASC (AS) = 68 THEN GOSUB 1090: GOTO 170
1205 IF Z < 0 THEN GOTO 1180
1210 IF XT > X OR XB < X OR YT > Y OR YB < Y THEN 1180
1220 T1 = X - 1: T2 = Y - 1: H = 0: XB = T1: YB = T2: GOSUB 1350
1230 GOSUB 1260: GOSUB 1310: IF Z < 0 THEN GOSUB 1330: GOTO 1190
1240 IF XT > X OR XB < X OR YT > Y OR YB < Y THEN 1230
1250 IF X < T1 OR Y < T2 THEN GOSUB 1350: PRINT DS;"PR#0": TEXT - HOME
    VTAB 12: HTAB 5: PRINT "PLEASE SPECIFY POINTS CORRECTLY": GOSUB 1300
    : GOSUB 1140: GOTO 1180
1260 WM = 1: WS = "": GOSUB 2590: RD = 0
1270 X3 = T1 + 1: Y3 = T2 + 1: X4 = X: Y4 = Y: H = 0: GOSUB 1350: GOSUB 1330: B
    1 = INT (X3 / 256): B2 = X3 - B1 * 256: B3 = INT ((X4 + 1) / 256): B4 =
    (X4 + 1) - B3 * 256: B5 = Y3: B6 = Y4 + 1: GOSUB 1120: HCOLOR= PC: GOTO
    170
1280 PRINT : PRINT DS;"PR#0": GOSUB 2300: PRINT DS;"PR#": SL: PRINT "M2": VTAB
    23: HTAB 15: POKE 41, PEEK (41) + 4: PRINT CHR$ (7): "LOWER-RIGHT?": FOR
    T3 = 1 TO 500: NEXT: PRINT DS;"PR#": SL: PRINT "N.H2": RETURN
1290 PRINT : PRINT DS;"PR#0": GOSUB 2300: PRINT DS;"PR#": SL: PRINT "M2": VTAB
    23: HTAB 15: POKE 41, PEEK (41) + 4: PRINT CHR$ (7): "UPPER-LEFT?": FOR
    T3 = 1 TO 500: NEXT: PRINT DS;"PR#": SL: PRINT "N.H2": RETURN
1300 FOR H = 1 TO 1000: NEXT H: RETURN
1310 PRINT DS;"IN#": SL: INPUT "": X, Y, Z: IF Z = 2 OR Z < 0 THEN RETURN
1320 GOTO 1310
1330 POKE 233, 99: POKE 232, 32: HCOLOR= 0: IF BC = 0 OR BC = 4 THEN HCOLOR=
    3
1340 H = 0: XB = X3 - 1: YB = Y3 - 1: GOSUB 1350: H = 16: XB = X4 + 1: YB = Y3 -
    1: GOSUB 1350: H = 32: XB = X4 + 1: YB = Y4 + 1: GOSUB 1350: H = 48: XB =
    X3 - 1: YB = Y4 + 1: GOSUB 1350: RETURN
1350 IF XB > = 0 AND XB < 280 AND YB > = 0 AND YB < 192 THEN ROT= H: SCALE=
    1: XDRAW 1 AT XB, YB
1360 RETURN
1380 RD = RD + 1: IF RD > 1 THEN RD = 0: GOSUB 1130: ON CM + 1 GOTO 170, 15
    80, 1680, 1740, 1840
1390 GOSUB 1400: ON CM + 1 GOTO 170, 1580, 1680, 1740, 1840
1400 IF X4 = X3 OR Y4 = Y3 THEN 1440
1410 T1 = ((XH * 2) - (XM * 2)) / (X4 - X3): T2 = ((YH * 2) - (YM * 2)) / (Y4 -
    Y3): SF = INT (T1): IF T2 < T1 THEN SF = INT (T2)
1420 XF = INT ((XM * 2) - (SF * X3)): YF = INT ((YM * 2) - (SF * Y3)): IF
    ABS (XF) > 27000 OR ABS (YF) > 27000 THEN GOTO 1440

```



```

1430 GOSUB 2590: RETURN
1440 PRINT DS;"PR#0": GOSUB 2300: PRINT : PRINT DS;"PR#": SL: PRINT "M2.C"
    : VTAB 23: HTAB 14: POKE 41, PEEK (41) + 4: PRINT "NOT POSSIBLE." RD =
    0: GOSUB 1300: GOSUB 1130: RETURN
1450 PRINT : PRINT DS;"PR#": SL: PRINT "T1.C": PRINT DS;"PR#0": PRINT DS;""
    IN#0": TEXT
1470 HOME: PRINT : HTAB 9: PRINT "FAST-DRAW DELTA SETTING": VTAB 5: HTAB
    7: PRINT "CURRENT DELTA SETTING IS": ABS (DX): "": PRINT : AS = "ON"
    : IF DX < 0 THEN AS = "OFF."
1480 HTAB 10: PRINT "AUDIO FEEDBACK IS": AS
1490 VTAB 18: CALL - 958: HTAB 11: INPUT "NEW DELTA EQUALS": AS: IF AS =
    "" THEN 1530
1500 IF VAL (AS) < 1 OR VAL (AS) > 127 THEN 1490
1510 IF DX < 1 THEN DX = - VAL (AS): GOTO 1530
1520 DX = VAL (AS)
1530 VTAB 20: CALL - 958: HTAB 9: INPUT "TURN AUDIO FEEDBACK": AS: IF AS
    = "" THEN 1560
1540 IF LEFT$ (AS, 2) < 0: "ON" AND LEFT$ (AS, 3) < 0: "OFF" THEN 1530
1550 DX = ABS (DX): IF LEFT$ (AS, 3) = "OFF" THEN DX = - DX
1560 GOSUB 1130: GOTO 170
1580 GOSUB 1130: PRINT DS;"IN#": SL: CM = 1
1590 RT = 2: INPUT X, Y, Z: IF Z < > 2 THEN POKE - 16368, 0: GOTO 1590
1600 IF X < X3 OR X > X4 OR Y < Y3 OR Y > Y4 THEN GOSUB 1940: IF RT = 1 THEN
    220
1610 IF RT = 0 THEN 1590
1620 HPLOT X, Y
1630 RT = 2: INPUT X, Y, Z: IF Z < > 2 THEN POKE - 16368, 0: GOTO 1630
1640 IF X < X3 OR X > X4 OR Y < Y3 OR Y > Y4 THEN GOSUB 1940: IF RT = 1 THEN
    220
1650 IF RT = 0 THEN 1630
1660 HPLOT TO X, Y: GOTO 1630
1680 GOSUB 1130: PRINT DS;"IN#": SL: CM = 2
1690 RT = 2: INPUT X, Y, Z: IF Z < > 2 THEN POKE - 16368, 0: GOTO 1690
1700 IF X < X3 OR X > X4 OR Y < Y3 OR Y > Y4 THEN GOSUB 1940: IF RT = 1 THEN
    220
1710 IF RT = 0 THEN 1690
1720 HPLOT X, Y: GOTO 1690
1740 GOSUB 1130: PRINT DS;"IN#": SL: CM = 3
1750 RT = 2: INPUT X, Y, Z: IF Z < > 2 THEN POKE - 16368, 0: GOTO 1750
1760 IF X < X3 OR X > X4 OR Y < Y3 OR Y > Y4 THEN GOSUB 1940: IF RT = 1 THEN
    220
1770 IF RT = 0 THEN 1750
1780 HPLOT X, Y: TX = X: TY = Y
1790 RT = 2: INPUT X, Y, Z: IF Z < > 2 THEN POKE - 16368, 0: GOTO 1790
1800 IF X < X3 OR X > X4 OR Y < Y3 OR Y > Y4 THEN GOSUB 1940: IF RT = 1 THEN
    220
1810 IF RT = 0 THEN 1790
1820 HPLOT X, Y TO TX, Y TO TX, TY TO X, TY TO X, Y: GOTO 1790
1840 GOSUB 1130: PRINT DS;"IN#": SL: CM = 4
1850 RT = 2: INPUT X, Y, Z: IF Z < > 2 THEN POKE - 16368, 0: GOTO 1850
1860 IF X < X3 OR X > X4 OR Y < Y3 OR Y > Y4 THEN GOSUB 1940: IF RT = 1 THEN
    220
1870 IF RT = 0 THEN 1850
1880 HPLOT X, Y: TX = X: TY = Y
1890 RT = 2: INPUT X, Y, Z: IF Z < > 2 THEN POKE - 16368, 0: GOTO 1890
1900 IF X < X3 OR X > X4 OR Y < Y3 OR Y > Y4 THEN GOSUB 1940: IF RT = 1 THEN
    220
1910 IF RT = 0 THEN 1890
1920 IF Y < TY THEN FOR H = Y TO TY: HPLOT X, H TO TX, H: NEXT : GOTO 1850
1930 FOR H = TY TO Y: HPLOT X, H TO TX, H: NEXT : GOTO 1850
1940 IF (Y < SF + YF - YL + 2) / 50 < 2 THEN RT = 1: RETURN
1950 PRINT DS;"PR#0": GOSUB 2300: PRINT DS;"PR#": SL: PRINT "M2": VTAB 23:
    HTAB 3: POKE 41, PEEK (41) + 4: PRINT "POINT OUTSIDE VIEWPORT. RESP
    ECIFY": GOSUB 1300: PRINT DS;"PR#": SL: PRINT "N.H2": RT = 0: RETURN
1970 GOSUB 1130: NX = 1: CALL EPX, CD = PEEK (7C01): ON CD + 1 GOTO 170, 197
    9, 1980, 1980
1975 IF NX = 1 THEN 1970
1980 HPLOT XX(1), YZ(1) TO XX(NX - 1), YZ(NX - 1): GOSUB 1990: GOSUB 1130: GOTO
    170
1990 PRINT DS;"PR#0": GOSUB 2300: PRINT DS;"PR#": SL: PRINT "M2": VTAB 23:
    HTAB 14: POKE 41, PEEK (41) + 4: PRINT CHR$ (7): "CALCULATING.": IF
    NX = 2 THEN AR = 0: GOTO 2020

```

```

2000 AR = 0: FOR T1 = 2 TO NX - 1: DX = X%(T1) - X%(T1 - 1): DY = Y%(T1) +
Y%(T1 - 1): Y / 2: AR = AR + DX + DY: NEXT T1
2010 AR = AR + X%(1) - X%(NX - 1) * ((Y%(1) + Y%(NX - 1)) / 2): AR = ABS
1AR) / WM / 2: IF AR < 999999999 THEN AR = (INT (AR * 100)) / 100
2020 GOSUB 2300: VTAB 23: BS = "AREA IS": POKE 41: PEEK (41) + 4: GOSUB 2
030 GOSUB 1300: GOSUB 1300: RETURN
2030 BS = BS + STR$ (AR) + " SQUARE " + WS + " "; HTAB 21 = INT (LEN (BS) / 2): PRINT BS: RETURN
2040 GOSUB 1130: NX = 1: CALL EPN: CD = PEEK (1700): ON CD + 1 GOTO 170, 207
5, 2080, 2090
2075 IF NX = 1 THEN 2070
2090 GOSUB 2090: GOSUB 1130: GOTO 170
2090 PRINT DS, "PR#0": GOSUB 2200: PRINT DS, "PR#": SL: PRINT "M2": VTAB 23
HTAB 14: POKE 41: PEEK (41) + 4: PRINT CHR$ (7): "CALCULATING...": IF
NX = 2 THEN DT = 0: GOTO 2110
2100 DT = 0: FOR T1 = 2 TO NX - 1: DX = X%(T1) - X%(T1 - 1): DY = Y%(T1) - Y
%(T1 - 1): DT = DT + SQR (DX * DX + DY * DY): NEXT : DT = DT / WM: IF
DT < 999999999 THEN DT = (INT (DT * 100)) / 100
2110 GOSUB 2300: VTAB 23: BS = "THE DISTANCE IS": POKE 41: PEEK (41) + 4
GOSUB 2120: GOSUB 1300: GOSUB 1300: RETURN
2120 BS = BS + STR$ (DT) + " " + WS + " "; HTAB 21 = INT (LEN (BS) / 2): PRINT
BS: RETURN
2130 GOSUB 1130
2140 GOSUB 2310
2150 PRINT DS, "IN#": SL: INPUT X, Y, Z: IF Z < 0 THEN PRINT DS, "IN#0": GET
AS: IF ASC (AS) = 13 THEN GOSUB 1130: GOTO 170
2160 IF Z < 0: Z THEN PRINT: GOTO 2180
2170 IF X3 < X OR X4 < X OR Y3 < Y OR Y4 < Y THEN 2170
2180 GOSUB 2320
2190 PRINT DS, "IN#": SL: INPUT TX, TY, Z: IF Z < 0 THEN PRINT DS, "IN#0": GET
AS: IF ASC (AS) = 13 THEN GOSUB 1130: GOTO 170
2200 IF Z < 0: Z THEN PRINT: GOTO 2220
2210 IF TX < X3 OR TX > X4 OR TY < Y3 OR TY > Y4 THEN 2210
2220 PRINT DS, "PR#": SL: PRINT "T1,C": TEXT : HOME : T1 = TX - X + 1: T2 = T
Y - Y + 1: DX = SQR (T1 * T1 + T2 * T2): VTAB 10: HTAB 6: PRINT "DIST
ANCE IS": INT (DX): " SCREEN UNITS": PRINT DS, "IN#0"
2230 VTAB 19: CALL - 958: HTAB 8: INPUT "YOUR NUMBER OF UNITS ->": AS: IF
AS = "" THEN W = DX: GOTO 2280
2240 IF VAL (AS) > 999999999 THEN 2260
2250 W = VAL (AS): IF W = 0 THEN 2260
2260 VTAB 20: CALL - 958: HTAB 8: INPUT "TYPE OF UNITS ->": WS: IF LEN
(WS) > 10 THEN 2280
2270 WM = DX / W: GOSUB 1130: GOTO 170
2280 FOR T4 = 21 TO 24: VTAB T4: HTAB 1: POKE 41: PEEK (41) + 4: PRINT "
": NEXT T4: PRINT : RETURN
2290 PRINT DS, "PR#0": GOSUB 2300: PRINT DS, "PR#": SL: PRINT "M2": VTAB 23
HTAB 13: POKE 41: PEEK (41) + 4: PRINT CHR$ (7): "BEGINNING POINT?": FOR
T3 = 1 TO 500: NEXT : PRINT DS, "PR#": SL: PRINT "N,H2": RETURN
2300 PRINT DS, "PR#0": GOSUB 2300: PRINT DS, "PR#": SL: PRINT "M2": VTAB 22
HTAB 14: POKE 41: PEEK (41) + 4: PRINT CHR$ (7): "ENDING POINT?": FOR
T3 = 1 TO 500: NEXT : PRINT DS, "PR#": SL: PRINT "N,H2": RETURN
2310 REM *** CHRIS' SLIDE ***
2320 GOSUB 1130: HCOLOR= BC: GOSUB 1040
2330 GOSUB 1130: GOSUB 2310
2340 PRINT DS, "IN#": SL: INPUT X, Y, Z: IF Z < 0 THEN PRINT DS, "IN#0": GET
AS: IF ASC (AS) = 13 THEN 2470
2350 IF Z < 0: Z THEN PRINT: GOTO 2360
2360 IF X < 0 OR X > 279 OR Y < 0 OR Y > 191 THEN 2350
2370 GOSUB 2320
2380 PRINT DS, "IN#": SL: INPUT TX, TY, Z: IF Z < 0 THEN PRINT DS, "IN#0": GET
AS: IF ASC (AS) = 13 THEN 2470
2390 IF Z < 0: Z THEN PRINT: GOTO 2400
2400 IF TX < 0 OR TX > 279 OR TY < 0 OR TY > 191 OR (TX = X AND TY = Y) THEN
2390
2410 IF TY > Y THEN FOR ZZ = 1 TO TY - Y: CALL 25218: NEXT: GOTO 2450
2420 IF Y > TY THEN FOR ZZ = 1 TO Y - TY: CALL 25175: NEXT
2430 IF TX > X THEN FOR ZZ = 1 TO INT ((TX - X) / 14): CALL 25308: NEXT
2440 IF X > TX THEN FOR ZZ = 1 TO INT ((X - TX) / 14): CALL 25261: NEXT
2450 HCOLOR= 0: IF BC = 0 OR BC = 4 THEN HCOLOR= 3
2460 GOSUB 1040: GOSUB 1330: HCOLOR= PC: GOSUB 1130: GOTO 170
2470 REM *** DAVE'S SEPERATE ***

```

```

2500 T3 = PC: GOSUB 670: IF PC = 4 OR PC = 0 THEN PC = T3: HCOLOR= PC: HOME
PRINT HTAB 9: PRINT "NO SEPARATION ON BLACKS.": GOSUB 1300: GOSUB
1410 GOSUB 1130: GOTO 170
2510 IF PC = 8 THEN PC = T3: GOTO 2580
2520 PRINT DS, "PR#": SL: PRINT "H2,N": GOSUB 1330: HCOLOR= BC: GOSUB 1040
IF PC = 3 OR PC = 7 THEN POKE 767, 176: CALL 24576: GOTO 2570
2530 POKE 767-144: CALL 24576: T1 = 128: T2 = 213: IF PC = 2 THEN T2 = 170
2540 IF PC = 5 THEN T1 = 0: T2 = 213
2550 IF PC = 6 THEN T1 = 0: T2 = 170
2560 POKE 768, T1: POKE 767, T2: CALL 24911
2570 BC = 0: HCOLOR= 3: GOSUB 1040: GOSUB 1330
2580 HCOLOR= PC: GOSUB 1130: GOTO 170
2590 PRINT : PRINT DS, "PR#": SL
2600 PRINT "D,S": SF: "H2,X": XF: "Y": YF: "R,N": RETURN
2610 TEXT : HOME : PRINT : HTAB 7: PRINT "TABLET INFORMATION FILE DOES": PRINT
HTAB 16: PRINT "NOT EXIST."
2620 VTAB 7: HTAB 8: PRINT "MAKE SURE THE MASTER DISK": PRINT HTAB 11: PRINT
"IS NOT PROTECTED AND": PRINT : HTAB 12: PRINT "THEN PRESS RETURN"
2630 VTAB 14: HTAB 5: PRINT "THE MENU ALIGNMENT ROUTINE WILL": PRINT : HTAB
17: PRINT "BE RUN.": GET AS: IF ASC (AS) < 13 THEN 2630
2635 POKE 104, 8: POKE 103, 1
2640 PRINT : PRINT DS, "RUN MENU ALIGNMENT.D1": STOP
2650 REM + ERROR HANDLER +
2660 TEXT : HOME : T7 = PEEK (222): PRINT DS, "PR#0": IF T7 = 8 THEN VTAB
12: HTAB 16: PRINT "I/O ERROR.": GOTO 2700
2670 IF T7 = 6 THEN VTAB 12: HTAB 11: PRINT "PICTURE NOT ON DISK.": GOTO
2700
2680 IF T7 = 4 OR T7 = 9 OR T7 = 10 THEN VTAB 12: HTAB 8: PRINT "THE PIC
TURE IS LOCKED, OR": HTAB 5: PRINT "THE DISK IS FULL, OR PROTECTED.": GOTO
2700
2685 IF T7 = 13 THEN VTAB 12: PRINT "FILE REQUESTED IS NOT A PICTURE FI
LE.": GOTO 2700
2690 VTAB 12: HTAB 9: PRINT "PROBLEM --> PEEK(222)=": T7
2700 VTAB 20: HTAB 8: PRINT "PRESS SPACE BAR TO RETRY.": PRINT : HTAB 11:
PRINT "PRESS CCR TO ABORT"
2710 VTAB 24: HTAB 20: GET AS: IF AS = " " THEN VTAB 20: HTAB 1: CALL -
958: HTAB 15: PRINT "TRYING.": IF T7 = 6 THEN GOTO 349
2715 IF AS = " " THEN RESUME
2720 IF ASC (AS) = 13 THEN PRINT : PRINT DS, "CLOSE": DS: GOTO 290
2730 GOTO 2710

```

VARIABLE ATLAS

Name	Description
AS	General-purpose input string
AR	Calculated area for AREA command
BS	Input string for picture name in LOAD and SAVE
BL-B6	Temporary variables for WINDOW
BC	Background Color (defaults to 0)
BX	Pointer into the QUICK-DRAW subroutines
CS	String for slot number
CD	Termination code of QUICK-DRAW subroutines
CM	Current Command mode: 0 = DRAW 1 = LINES 2 = DOTS 3 = FRAME 4 = BOX
DS	CTRL-D (CHR\$(4)) for DOS commands
DX	DELTA setting (0-127; negative if Audio Feedback is off)
DF	Default drive number for LOAD, SAVE, CATALOG

DT Calculated distance for DISTANCE command
 DX,DY Temporary variables used in AREA and DISTANCE: the vertical and horizontal distance between a point and the next one.
 ES Temporary input string for SAVE
 EP% The beginning address of the QUICK-DRAW subroutines
 GF X screen offset values for WINDOW
 H Widely used as a temporary variable.
 HF Y screen offset values for WINDOW
 LT Length of menu, in Tablet units
 MZ Maximum number of points for DISTANCE or AREA calculation
 MD Height and width of each menu command square, in Tablet units
 NZ Index into arrays X% and Y%, used by QUICK-DRAW
 PC Pen color (0-7), defaults to 3 (white)
 PI Number of points per inch on the Tablet
 RD Flag for REDUCER mode: 1=on, 0=off.
 RT A return flag for LINES, DOTS, FRAME, and BOX modes whose value indicates the phase of the operation:
 0 = Operation was just initialized.
 1 = Menu selected; operation cancelled.
 2 = Operation in progress.
 S0 Scale setting for menu
 S2 Scale setting for WINDOW after LOAD
 SF Scale Factor -- see XF,YF
 SL Slot number of Tablet Interface card (read from info file)
 TI-T9 Temporary variables
 TX,TY Temporarily holds an X,Y position (for BOX, LINES, FRAME, SLIDE)
 W User CALIBRATE units
 WS Name of user CALIBRATE units
 WM CALIBRATE multiplier (WM= Tablet units / W)
 X,Y General-purpose coordinate pair for high-resolution screen
 X%,Y% Arrays (of length MZ) which hold coordinates of points plotted in DRAW, AREA, and DISTANCE. They are filled by the QUICK-DRAW subroutines.
 X1,Y1 Coordinates for upper-left corner of WINDOW on Tablet
 X2,Y2 Coordinates for lower-right corner of WINDOW on Tablet
 X3,Y3 Coordinates for upper-left corner of VIEWPORT on screen
 X4,Y4 Coordinates for lower-right corner of VIEWPORT on screen
 X5,Y5 Default values for X1,Y1
 X6,Y6 Default values for X2,Y2
 X8,Y8 Temporary X,Y coordinates (for VIEWPORT and color menu)
 X9,Y9 " " " "
 XA,YA Width and height of menu overlay
 XB,YB Coordinates for lower-right corner of WINDOW on screen
 XF,YF Current Tablet offset factors
 XH,YH Coordinates for upper-left corner of overlay on Tablet
 XL,YL Coordinates for lower-right corner of overlay on Tablet
 XM,YM Coordinates for upper-left corner of working area on Tablet
 XT,YT Coordinates for upper-left corner of WINDOW on screen
 Z Pen up/pen down value:
 0 = pen is down, and has been down.
 1 = pen is up
 2 = pen newly down
 10 = pen is off-scale
 ZZ Negative numbers indicate that a key has been pressed.
 Temporary variable used in delay loops.

SUBROUTINES

<u>Entry</u>	<u>Description</u>
530	Inputs drive number from keyboard
670	Displays color menu; returns chosen color in PC
880	Draws a single box of the color C9 on the low-resolution graphics screen. The box will be X8 blocks tall, and its upper-left corner will be at (X9, Y9).
1040	Draws the WINDOW on the high-resolution screen in the current HCOLOR.
1070	Sets scaling information for Tablet; falls into subroutine at 1090
1090	Turns off REDUCER, removes WINDOW frame and sets WINDOW to its default values, resets CALIBRATE setting, and falls into subroutine at 1120
1120	Stores VIEWPORT setting in memory for QUICK-DRAW
1130	Resets Tablet scaling information (with REDUCER, if active)
1280	Prints prompt "LOWER-RIGHT?"
1290	Prints prompt "UPPER-LEFT?"
1300	Delay 1.1 seconds
1310	Wait for the pen to be down or a keypress. If pen is down, return with coordinates in X,Y; if keypress, return with Z<0.
1330	Draws or undraws the four VIEWPORT corner marks.
1350	Draws or undraws a single VIEWPORT corner mark. The corner's coordinates are in X8,Y8 and the rotation factor is in H.
1400	Turns on the REDUCER.
1940	Returns with RT=1 if the last pen press was in the menu area; otherwise displays "POINT OUTSIDE VIEWPORT. RESPECIFY"
1990	Performs an AREA calculation on the polygon whose vertices are in the arrays X%,Y%. Returns with the area in AR.
2090	Performs a DISTANCE calculation on the closed curve whose points are in the arrays X%,Y%. Returns with the distance in DT.
2120	Adds the value of DT to the end of string B\$, and prints it centered on the screen.
2300	Clears out the bottom four lines of the Page 2 Text screen.
2310	Displays the prompt "BEGINNING POINT?"
2320	Displays the prompt "ENDING POINT?"
2590	Reinitializes the Tablet with the scaling factor in SF, the X-offset in XF, and the Y-offset in YF.

SPECIAL LOCATIONS

These special memory locations are used by the TABLET-CODE APPLESOFT program. The decimal addresses are given on the left; hexadecimal equivalents are in parentheses and preceded by a dollar sign (\$):

<u>Location</u>	<u>Use</u>
41 (\$29)	This location contains the high part of the memory address of the beginning of the current line on the Text screen. A POKE 41, PEEK(41)+4 operation will cause the next printed line to appear on Page 2, rather than Page 1, of Text mode.
103,104 (\$67,\$68)	This pair of locations holds the address of the beginning of the current Applesoft program in memory.
222 (\$DE)	This location holds the ON ERR GOTO code of the last error generated.
232,233 (\$E8,\$E9)	This pair of locations holds the address of the beginning of the current shape table for the Applesoft DRAW and XDRAW commands.
700 (\$2BC)	Holds the termination code from the QUICK-DRAW subroutines.
752,753 (\$2F0,\$2F1)	After the QUICK-DRAW program is RUN, this pair of locations will hold the memory address of the beginning of the QUICK-DRAW subroutine.
766,767 (\$2FE,\$2FF)	These locations are used to pass the selected color to the SEPARATE subroutine.
3089-3094 (\$C11-\$C16)	These locations are used to pass VIEWPORT information to the QUICK-DRAW subroutine.
16632,16633 (\$40F8,\$40F9)	These locations are in the memory range used by the high-resolution graphics Page 2, but their contents are neither displayed on the screen or affected by normal screen operations. These two locations are used to store the value of S2 during a SAVE.
16504-16511 (\$4078-\$407F)	These are also locations in the high-resolution Page 2 which are not displayed. These eight locations are used to store the values of X1, X2, Y1, and Y2 during a SAVE.
24576 (\$6000)	This is the entry point for the machine language subroutine which performs a SEPARATE.

24911 (\$614F)	This is another entry point for SEPARATE.
25175 (\$6257)	Entry point for a one-dot SLIDE down.
25218 (\$6282)	Entry point for a one-dot SLIDE up.
25261 (\$62AD)	Entry point for a 14-dot SLIDE right.
25308 (\$62DC)	Entry point for a 14-dot SLIDE left.
-16368 (\$C010)	A PEEK or POKE to this location will clear the Apple's keyboard strobe, causing any recent keypress to be ignored.
62454 (\$F3F6)	This subroutine in the Applesoft ROM fills the entire high-resolution screen with the most recent HCOLOR plotted.
-958 (\$FC42)	This subroutine in the Apple's Monitor ROM clears the text screen from the current cursor position to the end of the screen.
-868 (\$FC9C)	This subroutine in the Apple's Monitor ROM clears the text screen from the current cursor position to the end of the line.

ROM CODE

```

SOURCE FILE: BITPAD35.1P
SOURCE FILE: BITPAD35.2P
0000: 1 ****
0000: 2 *
0000: 3 *
0000: 4 * BIT PAD FIRMWARE
0000: 5 *
0000: 6 * COPYRIGHT APPLE COMPUTER
0000: 7 * 7/30/79
0000: 8 * W SANDER
0000: 9 *
0000: 10 *
0000: 11 ****
0024: 12 CH EQU $24 ;SCREEN HORIZONTAL POSITION
002A: 13 HBASL EQU $24 ;BASE ADDRESS FOR BITPAD CURSOR
002B: 14 HDASH EQU $2B
002B: 15 BASL EQU $2B ;TEXT BASE ADDRESS
0036: 16 COUTL EQU $36 ;LOW BYTE OF COUT POINTER
0037: 17 COUTH EQU $37 ;HIGH BYTE OF COUT POINTER
0200: 18 INO EQU $200 ;INPUT BUFFER ADDRESSES
0201: 19 IN1 EQU $201
0202: 20 IN2 EQU $202
0203: 21 IN3 EQU $203
0280: 22 TEM EQU $280 ;RETURN FLAG LOCATION
0000: 23 * ;HIGH NIBBLE - 1=RETURN SCALED VALUE
0000: 24 * ;LOW NIBBLE - 0=OPEN DOWN, 1=PEN LIFT, 2=PEN FALL, 3=PEN UP
0281: 25 XFLL EQU $281 ;X-COORD LOW BYTE, FULL SCALE
0282: 26 XFLH EQU $282 ;X-COORD HIGH BYTE, FULL SCALE
0283: 27 YFLL EQU $283 ;Y-COORD LOW BYTE, FULL SCALE
0284: 28 YFLH EQU $284 ;Y-COORD HIGH BYTE, FULL SCALE
0285: 29 TEMXL EQU $285 ;X-COORD LOW BYTE, SCALED
0286: 30 TEMX EQU $286 ;X-COORD HIGH BYTE, SCALED

```

```

0287: 31 TEMYL EQU $287 : Y-COORD LOW BYTE, SCALED
0288: 32 TEMY EQU $288 : Y-COORD HIGH BYTE, SCALED
0287: 33 REGL EQU $287 : DIVIDE REGISTERS
0288: 34 REGH EQU $288
0290: 35 INA EQU $290 : BUFFER REGISTER FOR PR# SYNTAX
0298: 36 INX EQU $298 : PR# BUFFER POINTER
0299: 37 NFLAG EQU $299 : PR# BUFFER STATUS FLAG
0298: 38 SAVSLOT EQU $298
02A0: 39 MIFLAG EQU $2A0
02A1: 40 OREGL EQU $2A1
02A2: 41 OREGH EQU $2A2
02A3: 42 DIVL EQU $2A3
02A4: 43 DIVH EQU $2A4
02A5: 44 C1HAR EQU $2A5
0368: 45 PAGE EQU $388 : PAGE CODE
0000: 46 *
0000: 47 * HIGH BIT = 1 MEANS SCALE DATA
0000: 48 * 40 = HIRES PAGE2
0000: 49 * 20 = HIRES PAGE1
0000: 50 * 08 = TEXT PAGE2
0000: 51 * 04 = TEXT PAGE1
0000: 52 * 02 = LORES PAGE2
0000: 53 * 01 = LORES PAGE1
0000: 54 * 42 = HIRES MIXED PAGE2
0000: 55 * 21 = HIRES MIXED PAGE1
0000: 56 * 0A = LORES MIXED PAGE2
0000: 57 * 05 = LORES MIXED PAGE1
0000: 58 *
0438: 59 MPAGE EQU $438
0000: 60 *
0000: 61 * LAST SIX BITS OF MPAGE CORRESPOND TO PAGE
0000: 62 * BIT 7 MEANS STREAM MODE IF 1
0000: 63 * BIT 6 MEANS OFFSET AFTER SCALE IF 1
0000: 64 *
0488: 65 SCALL EQU $488 : LOW BYTE OF SCALE FACTOR
0538: 66 SCALH EQU $538 : HIGH BYTE OF SCALE FACTOR
0588: 67 OFFXL EQU $588 : LOW BYTE OF X-OFFSET
0638: 68 OFFXH EQU $638 : HIGH BYTE OF X-OFFSET
0688: 69 OFFYL EQU $688 : LOW BYTE OF Y-OFFSET
0738: 70 OFFYH EQU $738 : HIGH BYTE OF Y OFFSET
0678: 71 HNDX EQU $678 : TEMP INDEX FOR CURSOR PLOT
0578: 72 TEMPL EQU $578
05FB: 73 TEMPH EQU $5FB
02A5: 74 COUNT EQU $2A5 : UTILITY COUNT REG
06FB: 75 CHAR EQU $6FB : TEMPORARY CHARACTER STORE
07FB: 76 MSLOT EQU $7FB : CURRENT SLOT POINTER $CN
0000: 77 KBD EQU $C000 : KEYBOARD STROBE
C010: 78 KBDSTRB EQU $C010 : KEYBOARD STROBE RESET
C050: 79 SGR EQU $C050 : DISPLAY MODE REFERENCES
C051: 80 STEXT EQU $C051
C052: 81 SNMIX EQU $C052
C053: 82 SMIX EQU $C053
C054: 83 SPAG1 EQU $C054
C055: 84 SPAG2 EQU $C055
C056: 85 SLDRES EQU $C056
C057: 86 SHIRES EQU $C057
C061: 87 DEVO EQU $C081 : BITPAD DEVICE ADDRESSES
C080: 88 DEV1 EQU $C080
C083: 89 DEV2 EQU $C083
C082: 90 DEV3 EQU $C082
CFFF: 91 ROMSW EQU $CFFF : REFERENCE ADDRESS TO FREE $C800
FE93: 92 SETVID EQU $FE93 : SET CHARACTER OUTPUT TO NORMAL
FD0D: 93 COUT EQU $FD0D : CHARACTER OUTPUT
FF58: 94 IDRTS EQU $FF58 : UTILITY LOCATION CONTAINING RTS
0000: 95 *****
0000: 96 *****
0000: 97 *
0000: 98 * CNO0 ROM ENTRY
0000: 99 * FLAG SET-UP: C CLEAR FOR IN# ENTRY
0000: 100 * C SET FOR POINT RETURN (CURSOR)
0000: 101 *
0000: 102 * CNO0 ENTRY CODE REPLICATED FOR EACH
0000: 103 * N FROM 9 TO F (CORRESPONDING TO 1 TO 7)
0000: 104 *
0000: 105 *****

```

```

0000: 0000: 107 *****
0000: 0000: 108 *
0000: 0000: 109 * C800 SPACE ENTRY
0000: 0000: 110 *
0000: 0000: 111 * GET SLOT NO, SAVE MSLOT,
0000: 0000: 112 * AND SET DEFAULTS
0000: 0000: 113 *
0000: 0000: 114 *****
0000: 0000: ---- NEXT OBJECT FILE NAME IS XX
CB00: 115 ORG $C800
0000: 0000: 116 OBJ $6000
CB00: BD A5 02 117 OTHROM STA C1HAR : SAVE ACCUM FOR PR# ROUTINE
CB03: 68 118 PLA : PULL RETURN VECTOR TO
CB04: 68 119 PLA : GET SLOT NO.
CB05: 28 120 PLP
CB06: 50 01 121 DVC AOTHROM
CB08: 60 122 RTS
CB09: BD FB 07 123 AOTHROM STA MSLOT : SAVE SLOT NO.
CB0C: BD 9B 02 124 STA SAVSLOT
CB0F: AD A5 02 125 LDA C1HAR
CB12: BD FB 06 126 STA CHAR
CB15: 48 127 PHA : SAVE ACCUM
CB16: 8A 128 TXA : SAVE X-REG AND Y-REG
CB17: 48 129 PHA
CB18: 98 130 TYA
CB19: 48 131 PHA
CB1A: 08 132 PHP : SAVE STATUS
CB1B: AE FB 07 133 LDX MSLOT : LOAD X FOR SLOT DEP VARS
CB1E: BD 38 04 134 LDA MPAGE, X
CB21: 49 25 135 EOR #$25
CB23: 5D 38 03 136 EOR PAGE, X
CB26: 29 3F 137 AND #$3F
CB28: F0 03 138 BEQ PRCHK : IF SO THEN NO DEFAULT
CB2A: 20 90 CE 139 JSR DEFAULT
CB2D: E4 37 140 PRCHK CPX COUTH : CHECK IF FROM PR#
CB2F: D0 03 141 BNE NOPR
CB31: 4C AD CC 142 JMP SYNTAX : IF SO, THEN TAKE IN COMMANDS
CB34: 28 143 NOPR PLP : RECOVER STATUS
CB35: 08 144 PHP : AND SAVE
CB36: 80 07 145 BCS EPPOINT
CB38: AD FB 06 146 LDA CHAR
CB3B: A4 24 147 LDY CH
CB3D: 91 28 148 STA (BASL), Y : ELIM FLASHING CURSOR
CB3F: 150 *****
CB3F: 151 *****
CB3F: 152 *
CB3F: 153 * MAIN LOOP ENTRY
CB3F: 154 *
CB3F: 155 *****
CB3F: 156 *****
CB3F: 20 B9 CB 157 EPPOINT JSR MREAD : READ BITPAD
CB42: 2C 00 C0 158 BIT KBD
CB45: 30 26 159 BMI END
CB47: AD 80 02 160 LDA TEM : CHECK IF PEN DOWN
CB4A: 29 03 161 AND #$3
CB4C: C9 03 162 CMP #$3
CB4E: D0 1D 163 BNE END : EXIT IF PEN DOWN
CB50: AC FB 07 164 LDY MSLOT
CB53: B9 BB 03 165 LDA PAGE, Y
CB56: 29 7F 166 AND #$7F
CB58: F0 0E 167 BEQ E1PNT
CB5A: 20 F0 CB 168 JSR CURSOUT : DRAW CURSOR
CB5D: A9 60 169 LDA #$60
CB5F: 20 A1 CC 170 JSR QWAIT : LEAVE CURSOR ON FOR AWHILE
CB62: 20 F0 CB 171 JSR CURSOUT : THEN DELETE CURSOR
CB65: AC FB 07 172 LDY MSLOT
CB68: B9 38 04 173 E1PNT LDA MPAGE, Y
CB6B: 10 D2 174 SPL EPPOINT

```

```

C86D: 176 *****
C86D: 177 *****
C86D: 178 *
C86D: 179 * EXIT ROUTINES
C86D: 180 *
C86D: 181 *****
C86D: 182 *****
C86D: A2 03 183 END LDX #$3
C86F: BD 81 02 184 INEX1 LDA XFLL,X : COPY RESULTS TO TEM REGISTERS
C872: 9D 85 02 185 STA TEMXL,X
C875: CA 186 DEX
C876: 10 F7 187 BPL INEX1
C878: AE FB 07 188 LDX MSLOT : TEST HIGH BIT OF 'PAGE' TO
C87B: BD B8 03 189 LDA PAGE,X : SEE IF DATA IS TO BE SCALED
C87E: 10 03 190 SPL INEX2
C880: 20 70 CB 191 JSR SCALE : SCALE AND OFFSET DATA INTO TEM-X
C883: 28 192 INEX2 PLP
C884: 90 05 193 BCC INEXIT : SKIP TO FURTHER PROCESSING
C886: 68 194 EXIT PLA
C887: A8 195 TAY
C888: 68 196 PLA : RESTORE REGISTERS AND EXIT
C889: AA 197 TAX : IF NOT FROM INH#
C88A: 68 198 PLA
C88B: 60 199 RTS
C88C: 200 *
C88C: 201 * SET INPUT BUFFER TO +0000, +0000, +00
C88C: 202 *
C88C: 203 *****
C88C: A9 80 204 INEXIT LDA #$80
C88E: A0 0E 205 LDY #$E
C890: 99 00 02 206 QLOOP STA INO,Y
C893: 88 207 DEY
C894: 10 FA 208 BPL QLOOP
C896: A9 AB 209 LDA #$AB
C898: BD 00 02 210 STA $200
C898: BD 06 02 211 STA $206
C89E: BD 0C 02 212 STA $20C
C8A1: A9 AC 213 LDA #$AC
C8A3: BD 05 02 214 STA $205
C8A6: BD 0B 02 215 STA $20B
C8A7: AD 80 02 216 LDA TEM
C8AC: 29 10 217 AND #$10
C8AE: D0 16 218 BNE ASCEX
C8B0: AD 85 02 219 LDA TEMXL
C8B3: AC 86 02 220 LDY TEMX : CONVERT X TO ASCII
C8B6: A2 00 221 LDX #$0 : IN INPUT BUFFER
C8B8: 20 6A CA 222 JSR ASCON
C8B8: AD 87 02 223 LDA TEMYL
C8BE: AC 88 02 224 LDY TEMY
C8C1: A2 06 225 LDX #$6 : CONVERT Y TO ASCII
C8C3: 20 6A CA 226 JSR ASCON : IN INPUT BUFFER
C8C6: 2C 00 C0 227 ASCEX BIT KBD
C8C9: 10 09 228 BPL ASC1EX
C8CB: A9 AD 229 LDA #$AD
C8CD: BD 0C 02 230 STA $20C
C8D0: AD 80 02 231 ASC1EX LDA TEM
C8D3: 48 232 PHA
C8D4: 29 OF 233 AND #$OF : SET UP RETURN FLAG IN INPUT BUFF
C8D6: 09 BD 234 DRA #$80
C8D8: BD 0E 02 235 STA $20E
C8DB: 68 236 PLA
C8DC: 4A 237 LSR A
C8DD: 4A 238 LSR A
C8DE: 4A 239 LSR A
C8DF: 4A 240 LSR A
C8E0: 29 OF 241 AND #$OF
C8E2: 09 BD 242 DRA #$80
C8E4: BD 0D 02 243 STA $20D
C8E7: A8 244 TAY
C8E8: 68 245 PLA : PULL STACK AND SET UP
C8E9: 68 246 PLA : REGISTERS FOR END OF LINE
C8EA: 68 247 PLA
C8EB: A2 OF 248 LDX #$OF
C8ED: A7 BD 249 LDA #$BD
C8EF: 60 250 RTS

```

```

C8F0: 252 *****
C8F0: 253 *****
C8F0: 254 *
C8F0: 255 * CURSOR ROUTINE
C8F0: 256 *
C8F0: 257 *****
C8F0: 258 *****
C8F0: 20 70 CB 259 CURSOUT JSR SCALE
C8F3: 4C 0C C9 260 JMP C1SKP
C900: 261 ORG $100+OTHROM
C900: 18 262 INPUTXY CLC : CARRY SET FOR INH ENTRY
C901: 80 263 DFB $80 : SKIP NEXT BYTE
C902: 38 264 POINT SEC : SET CARRY FOR ENTRY TO GET A POI
C903: 88 265 CLV
C904: 08 266 PHP : SAVE FLAGS FOR LATER
C905: 78 267 SEI : DISABLE INTERRUPT UNTIL MSLOT SET
C906: 2C FF CF 268 BIT ROMSH : SWITCH OFF ALL $C800 ROMS
C909: 20 00 CB 269 JSR OTHROM : SWITCH TO $C800 SPACE
C90C: A5 2A 270 C1SKP LDA HBASL
C90E: 48 271 PHA
C90F: A5 2B 272 LDA HBASH
C911: 48 273 PHA
C912: AD 87 02 274 LDA TEMYL
C915: AE 85 02 275 LDX TEMXL
C918: 20 4C CA 276 CALLCURS JSR WINCHK
C91B: 48 277 PHA
C91C: B0 30 278 BCS OUT1
C91E: 20 F0 C9 279 JSR BASCLC
C921: 280 *****
C921: 281 *
C921: 282 * MODE EVALUATION
C921: 283 *
C921: 284 *****
C921: B9 88 03 285 LDA PAGE,Y
C924: 29 7F 286 AND #$7F
C926: A8 287 TAY
C927: 29 0C 288 AND #$80C
C929: F0 26 289 BEG CR
C92B: 98 290 TYA
C92C: 29 63 291 AND #$63
C92E: F0 06 292 BEG TEXT
C930: 68 293 MIX PLA : MIXED GRAPHICS BOUNDARY TEST
C931: 48 294 PHA
C932: C9 A0 295 CMP #160
C934: 90 18 296 BCC GR
C936: 297 *****
C936: 298 *
C936: 299 * TEXT MODE CURSOR GENERATION
C936: 300 *
C936: 301 *****
C936: 98 302 TEXT TYA
C937: 20 DE C9 303 JSR LOCLC
C93A: B1 2A 304 LDA (HBASL),Y
C93C: 48 305 PHA
C93D: A9 DF 306 LDA #$DF
C93F: 91 2A 307 STA (HBASL),Y
C941: A9 80 308 LDA #$80
C943: 20 A1 CC 309 JSR GWAIT
C946: 68 310 PLA
C947: 91 2A 311 STA (HBASL),Y
C949: 49 60 312 LDA #$60
C94B: 20 A1 CC 313 JSR GWAIT
C94E: 4C 00 C9 314 OUT1 JMP OUT
C951: 98 315 GR TYA : TEST FOR HIRES VS LORES GRAPHICS
C952: C9 OF 316 CMP #$OF
C954: B0 1B 317 BCS HIRES
C956: A5 2B 318 LDA HBASH
C958: 29 10 319 AND #$10
C95A: 320 *****
C95A: 321 *
C95A: 322 * LD RESOLUTION GRAPHICS
C95A: 323 * CURSOR ROUTINE
C95A: 324 *
C95A: 325 *****

```

C95A: 08 326 LORES PHP
 C95B: A9 F0 327 LDA #\$FO
 C95D: 28 328 PLP
 C95E: D0 02 329 BNE LOR1
 C960: 49 FF 330 EOR #\$FF
 C962: 48 331 LOR1 PHA
 C963: 98 332 TYA
 C964: 0A 333 ASL A
 C965: 0A 334 ASL A
 C966: 20 DE C9 335 JSR LOCLC
 C969: 68 336 PLA
 C96A: 51 2A 337 EOR (HBASL), Y
 C96C: 91 2A 338 STA (HBASL), Y
 C96E: 4C D0 C9 339 JMP OUT
 C971: 340 *****
 C971: 341 *
 C971: 342 * HIRES CURSOR ROUTINE
 C971: 343 *
 C971: 344 *****
 C971: A9 16 345 HIRES LDA #\$16
 C973: 8D A5 02 346 STA COUNT
 C976: 8A 347 TXA
 C977: 38 348 SEC
 C978: E9 05 349 SBC #\$5
 C97A: AA 350 TAX
 C97B: B0 03 351 BCS LOOP
 C97D: CE 86 02 352 DEC TEMX
 C980: A0 05 353 LOOP LDY #\$5
 C982: AD A5 02 354 LDA COUNT
 C985: D9 DB C9 355 CTRLOOP CMP CTRCHK, Y
 C988: F0 14 356 BEQ OUTSIDE
 C98A: BB 357 DEY
 C98B: 10 F8 358 BPL CTRLOOP
 C98D: 68 359 PLA
 C98E: 20 4C CA 360 JSR WINCHK
 C991: 48 361 PHA
 C992: B0 0A 362 BCS OUTSIDE
 C994: 20 F0 C9 363 JSR BASCLC
 C997: AC 78 06 364 LDY HNDX
 C99A: 51 2A 365 EOR (HBASL), Y
 C99C: 91 2A 366 STA (HBASL), Y
 C99E: A9 DC 367 OUTSIDE LDA #\$C
 C9A0: CD A5 02 368 CMP COUNT
 C9A3: F0 0E 369 BEQ A
 C9A5: B0 1D 370 BCS B
 C9A7: E8 371 INX
 C9A8: D0 03 372 BNE C
 C9AA: EE 86 02 373 INC TEMX
 C9AD: CE A5 02 374 C DEC COUNT
 C9B0: 4C B0 C9 375 JMP LOOP
 C9B3: 8A 376 A TXA
 C9B4: E9 05 377 SBC #\$5
 C9B6: AA 378 TAX
 C9B7: AD 86 02 379 LDA TEMX
 C9B9: E9 00 380 SBC #\$0
 C9Bc: 8D 86 02 381 STA TEMX
 C9Bf: 68 382 PLA
 C9C0: 38 383 SEC
 C9C1: E9 06 384 SBC #\$6
 C9C3: 48 385 PHA
 C9C4: 68 386 B PLA
 C9C5: 18 387 CLC
 C9C6: 69 01 388 ADC #\$1
 C9C8: 48 389 PHA
 C9C9: CE A5 02 390 DEC COUNT
 C9CC: F0 02 391 BEQ OUT
 C9CE: D0 B0 392 BNE LOOP
 C9D0: 68 393 OUT PLA
 C9D1: 68 394 PLA
 C9D2: 85 28 395 STA HBASH
 C9D4: 68 396 PLA
 C9D5: 85 2A 397 STA HBASL
 C9D7: 60 398 RTS

C9D8: 12 399 CTRCHK DFB \$12
 C9D9: 11 400 DFB \$11
 C9DA: 10 401 DFB \$10
 C9DB: 07 402 DFB \$7
 C9DC: 04 403 DFB \$6
 C9DD: 05 404 DFB \$5
 C9DE: 405 *****
 C9DE: 406 * TEXT BASE CALC
 C9DE: 407 *
 C9DE: 408 * THIS SUBROUTINE SETS UP BASE REGISTER
 C9DE: 409 * FOR LORES OR TEXT
 C9DE: 410 * ENTER WITH 'PAGE' IN A --
 C9DE: 411 * EXIT WITH HNDX IN Y. READY FOR SCREEN PROCESSING
 C9DE: 412 *
 C9DE: 413 *****
 C9DE: 414 LOCLC PHA
 C9DF: A5 28 415 LDA HBASH
 C9E1: 29 03 416 AND #\$3
 C9E3: B5 28 417 STA HBASH
 C9E5: 68 418 PLA
 C9E6: 29 0C 419 AND #\$0C
 C9E8: 05 28 420 ORA HBASH
 C9EA: B5 28 421 STA HBASH
 C9EC: AC 78 06 422 LDY HNDX
 C9EF: 60 423 RTS
 C9FO: 425 *****
 C9FO: 426 *
 C9FO: 427 * SCREEN BASE ADDRESS CALC
 C9FO: 428 *
 C9FO: 429 * ENTER WITH LO BYTE OF Y IN ACCUM
 C9FO: 430 * AND WITH LO BYTE OF X IN X-REG
 C9FO: 431 *
 C9FO: 432 * BASE ADDRESS WILL BE COMPUTED INTO
 C9FO: 433 * HBASL, HBASH, AND HNDX: HIRES BIT ADDRESSED
 C9FO: 434 * IS RETURNED AS A 1 IN A BIT IN THE ACCUM
 C9FO: 435 *
 C9FO: 436 *****
 C9FO: 437 BASCLC PHA
 C9F1: 29 C0 438 AND #\$C0
 C9F3: B5 2A 439 STA HBASL
 C9F5: 4A 440 LSR A
 C9F6: 4A 441 LSR A
 C9F7: 05 2A 442 ORA HBASL
 C9F9: B5 2A 443 STA HBASL
 C9FB: 68 444 PLA
 C9FC: 4C 0C CA 445 JMP C2SKP
 CA00: 446 ORG \$200+0THROM
 CA00: 18 447 CLC \$200 SPACE ENTRY
 CA01: B0 448 DFB \$B0
 CA02: 38 449 SEC
 CA03: B8 450 CLV
 CA04: 08 451 PHP
 CA05: 78 452 SEI
 CA06: 2C FF CF 453 BIT ROM\$4
 CA09: 20 00 CB 454 JSR 0THROM
 CA0C: B5 28 455 C2SKP STA HBASH
 CA0E: 0A 456 ASL A
 CA0F: 0A 457 ASL A
 CA10: 0A 458 ASL A
 CA11: 26 28 459 ROL HBASH
 CA13: 0A 460 ASL A
 CA14: 26 28 461 ROL HBASH
 CA16: 0A 462 ASL A
 CA17: 66 2A 463 ROR HBASL
 CA19: A5 28 464 LDA HBASH
 CA1B: 29 1F 465 AND #\$1F
 CA1D: B5 28 466 STA HBASH
 CA1F: 8A 467 TXA
 CA20: AC 86 02 468 LDY TEMX
 CA23: C0 00 469 CPY #\$0
 CA25: F0 05 470 BEQ HPOSN2
 CA27: A0 23 471 LDY #\$23
 CA29: 69 04 472 ADC #\$4

CA23: C8 473 HPOSN1 INY
 CA20: E9 07 474 HPOSN2 BCS #E7
 CA2E: B0 FB 475 BCS HPOSN1
 CA30: 69 08 476 ADC #E8
 CA32: BC 78 06 477 STY HNDX
 CA35: AB 478 TAY
 CA36: A9 00 479 LDA #E0
 CA3B: 38 480 SEC
 CA39: 2A 481 CLOOP ROL A
 CA3A: 88 482 DEY
 CA3B: D0 FC 483 BNE CLOOP
 CA3D: AC FB 07 484 LDY MSLOT
 CA40: 48 485 PHA
 CA41: B9 BB 03 486 LDA PAGE, Y
 CA44: 29 60 487 AND #E60
 CA46: 05 28 488 ORA HBASH
 CA48: 85 28 489 STA HBASH
 CA4A: 68 490 PLA
 CA4B: 60 491 RTS
 CA4C: 492 CHN BITPAD35, 2P
 CA4C: 2 *****
 CA4C: 3 *
 CA4C: 4 * WINDOW CHECK SUBROUTINE
 CA4C: 5 *
 CA4C: 6 * ENTER WITH LO BYTE OF Y IN ACCUM
 CA4C: 7 * AND LO BYTE OF X IN X-REG
 CA4C: 8 *
 CA4C: 9 * RETURN WILL BE CARRY CLEAR IF
 CA4C: 10 * WITHIN WINDOW AND CARRY SET IF
 CA4C: 11 * OUTSIDE.
 CA4C: 12 *
 CA4C: 13 *****
 CA4C: 14 WINCHK PHA
 CA4D: C9 C0 15 CMP #192
 CA4F: B0 16 16 BCS NO
 CA51: AD 86 02 17 LDA TEMX
 CA54: F0 09 18 BEQ YES
 CA56: C9 02 19 CMP #E2
 CA58: B0 00 20 BCS NO
 CASA: 8A 21 TXA
 CA5B: C9 18 22 CMP #24
 CA5D: B0 08 23 BCS NO
 CA5F: AD 88 02 24 YES LDA TEMY
 CA62: D0 03 25 BNE NO
 CA64: 18 26 CLC
 CA65: 68 27 PLA
 CA66: 60 28 RTS
 CA67: 38 29 NO SEC
 CA68: 68 30 PLA
 CA69: 60 31 RTS
 CA6A: 33 *****
 CA6A: 34 *
 CA6A: 35 * ASCII CONVERSION
 CA6A: 36 * ENTER WITH HIGH BYTE
 CA6A: 37 * IN Y-REG, LOW BYTE IN ACCUM
 CA6A: 38 * AND INPUT BUFFER OFFSET
 CA6A: 39 * IN X-REG
 CA6A: 40 *
 CA6A: 41 *****
 CA6A: 42 ASCON PHA
 CA6B: 98 43 TYA
 CA6C: 10 12 44 BPL POSIT
 CA6E: 48 45 PHA
 CA6F: A9 AD 46 LDA #EAD
 CA71: 9D 00 02 47 STA INO, X
 CA74: 68 48 PLA
 CA75: 86 2A 49 STX HBASL
 CA77: AA 50 TAX
 CA78: 68 51 PLA
 CA79: 20 58 CB 52 JSR TWOCON
 CA7C: 48 53 PHA
 CA7D: 8A 54 TXA
 CA7E: A6 2A 55 LDX HBASL

CAB0: AB 56 POSIT TAY
 CAB1: EB 57 INX
 CAB2: 68 58 PLA
 CAB3: 38 59 SEC
 CAB4: B0 03 60 BCS ASKIP
 CAB6: FE 00 02 61 FLOOP INC INO, X
 CAB9: E9 EB 62 ASKIP BSC #E8
 CAB8: 48 63 PHA
 CABC: 98 64 TYA
 CABD: E9 03 65 SBC #E3
 CABF: AB 66 TAY
 CA90: 68 67 PLA
 CA91: B0 F3 68 BCS FLOOP
 CA93: 69 EB 69 ADC #E8
 CA95: 48 70 PHA
 CA96: 98 71 TYA
 CA97: 69 03 72 ADC #E3
 CA99: AB 73 TAY
 CA9A: 68 74 PLA
 CAB8: B0 03 75 BCS BSKIP
 CABD: FE 01 02 76 HLOOP INC IN1, X
 CAA0: 38 77 BSKIP SEC
 CAA1: E9 64 78 SBC #E4
 CAA3: B0 FB 79 BCS HLOOP
 CAA5: 98 80 DEY
 CAA6: 10 F5 81 SPL HLOOP
 CAA8: 18 82 CLC
 CAA9: C8 83 INY
 CAAA: 69 64 84 ADC #E4
 CAAC: 38 85 SEC
 CAAD: B0 03 86 BCS CSKIP
 CAAF: FE 02 02 87 JLOOP INC IN2, X
 CAB2: E9 0A 88 CSKIP SBC #E4
 CAB4: B0 F9 89 BCS JLOOP
 CAB6: 69 0A 90 ADC #E4
 CAB8: B0 03 91 BCS DSKIP
 CABA: FE 03 02 92 KLOOP INC IN3, X
 CABD: E9 01 93 DSKIP SBC #E1
 CABF: B0 F9 94 BCS KLOOP
 CAC1: 60 95 RTN RTS
 CAC2: 97 *****
 CAC2: 98 *
 CAC2: 99 * OFFSET AND DIVIDE ROUTINE
 CAC2: 100 *
 CAC2: 101 * ENTER WITH OFFSET IN REGH
 CAC2: 102 * AND REGL, VALUE TO SCALE IN
 CAC2: 103 * X (HIGH BYTE) AND A (LOW BYTE)
 CAC2: 104 * SCALE (INTEGER DIVISOR) IN
 CAC2: 105 * DIVH AND DIVL
 CAC2: 106 *
 CAC2: 107 * RESULT IN REGH AND REGL
 CAC2: 108 * VALUE MAY BE +OR-32767
 CAC2: 109 * SCALE 0 TO +32767
 CAC2: 110 * RESULT IS TWOS COMPLEMENT
 CAC2: 111 * REMAINDER IS LOST
 CAC2: 112 *
 CAC2: 113 *****
 CAC2: AC FB 07 114 OFFDIV LDY MSLOT
 CAC5: 48 115 PHA
 CAC6: B9 BB 04 116 LDA MPAGE, Y
 CAC9: 0A 117 ASL A
 CACA: 30 05 118 BMI DIVIDE
 CACC: 68 119 PLA
 CACD: 20 64 CB 120 JSR OFFSET
 CAD0: 48 121 PHA
 CAD1: 88 122 DIVIDE CLV
 CAD2: A0 00 123 LDY #E0
 CAD4: AD A4 02 124 LDA DIVL
 CAD7: C8 125 LOOP1 INY
 CADB: 0E A3 02 126 ASL DIVL
 CADB: 2A 127 ROL A
 CADC: 10 F9 128 SPL LOOP1
 CADF: BD A4 02 129 STA DIVH

CAE1: A9 00	130	LDA	#\$0
CAE3: BD BB 02	131	STA	REQH
CAE6: BD B7 02	132	STA	REGL
CAE9: 68	133	PLA	
CAEA: EB	134	INX	
CAEB: CA	135	DEX	
CAEC: 08	136	PHP	
CAED: 10 03	137	BPL	POS
CAEF: 20 58 CB	138	JSR	TWOCOM
CAF2: 38	139	POS	SEC
CAF3: ED A3 02	140	LOOP2	SBC
CAF6: 48	141	PHA	
CAF7: 8A	142	TXA	
CAF8: ED A4 02	143	SBC	DIVH
CAF9: AA	144	TAX	
CAF0: 68	145	PLA	
CAF0: 4C 0C CB	146	JMP	LOOP3
CB00: 18	147	ORG	\$300+OTHROM
CB00: 18	148	CLC	
CB01: 80	149	DFB	\$80
CB02: 38	150	SEC	
CB03: 88	151	CLV	
CB04: 08	152	PHP	
CB05: 78	153	SEI	
CB06: 2C FF CF	154	BIT	ROMSW
CB09: 20 00 CB	155	JSR	OTHROM
CB0C: 08	156	LOOP3	PHP
CB0D: 2E B7 02	157	ROL	REGL
CB10: 2E 88 02	158	ROL	REQH
CB13: 88	159	DEY	
CB14: 30 16	160	BMI	FEXIT
CB16: 4E A4 02	161	LSR	DIVH
CB19: 6E A3 02	162	ROR	DIVL
CB1C: 28	163	PLP	
CB1D: 80 D4	164	BCS	LOOP2
CB1F: BD A3 02	165	LOOP4	ADC
CB22: 48	166	PHA	DIVL
CB23: 8A	167	TXA	
CB24: 6D A4 02	168	ADC	DIVH
CB27: AA	169	TAX	
CB28: 68	170	PLA	
CB29: 4C 0C CB	171	JMP	LOOP3
CB2C: 28	172	FEXIT	PLP
CB2D: 28	173	DEXIT	PLP
CB2E: 10 0F	174	BPL	EEXIT
CB30: AD B7 02	175	LDA	REGL
CB33: AE 88 02	176	LDX	REQH
CB36: 20 58 CB	177	JSR	TWOCOM
CB39: 8E BB 02	178	STX	REQH
CB3C: BD B7 02	179	STA	REGL
CB3F: AC FB 07	180	EEXIT	LDY
CB42: B9 38 04	181	LDA	MPAGE, Y
CB45: 0A	182	ASL	A
CB46: 10 0F	183	BPL	E1EXIT
CB48: AD B7 02	184	LDA	REGL
CB4B: AE 88 02	185	LDX	REQH
CB4E: 20 64 CB	186	JSR	OFFSET
CB51: BD B7 02	187	STA	REGL
CB54: 8E BB 02	188	STX	REQH
CB57: 60	189	E1EXIT	RTS
CB58: 49 FF	190	TWOCOM	EOR
CB5A: 18	191	CLC	#\$FF
CB5B: 69 01	192	ADC	#\$1
CB5D: 48	193	PHA	
CB5E: 8A	194	TXA	
CB5F: 49 FF	195	EOR	#\$FF
CB61: AA	196	TAX	
CB62: 68	197	PLA	
CB63: 60	198	RTS	
CB64: 38	199	OFFSET	SEC
CB65: ED A1 02	200	SBC	OREGL
CB68: 48	201	PHA	
CB69: 8A	202	TXA	

CB6A: ED A2 02	203	SBC	OREGH
CB6D: AA	204	TAX	
CB6E: 68	205	PLA	
CB6F: 60	206	RTS	
CB70:	208	*****	
CB70:	209	*	
CB70:	210	*	SCALE ROUTINE
CB70:	211	*	
CB70:	212	*	VALUES IN -FL- REGISTERS CONVERTED
CB70:	213	*	TO SCALED VALUES IN TEM- REGISTERS
CB70:	214	*	
CB70:	215	*	*****
CB70:	216	SCALE	LDY
CB73: B9 BB 04	217	MSLOT	LDA
CB76: BD A3 02	218	SCALL, Y	STB
CB79: B9 38 05	219	SCALH, Y	LDA
CB7C: BD A4 02	220	DIVH	STB
CB7F: B9 BB 05	221	OFFXL, Y	LDA
CB82: BD A1 02	222	OREGL	STB
CB85: B9 38 06	223	OFFXH, Y	LDA
CB88: BD A2 02	224	OREGH	STB
CB8B: AD 91 02	225	XFL	LDA
CB8E: AE 62 02	226	XFLH	LDX
CB91: 20 C2 CA	227	OFFDIV	JSR
CB94: AC FB 07	228	MSLOT	LDY
CB97: AD 97 02	229	TEMYL	LDA
CB9A: BD B5 02	230	TEMXL	STB
CB9D: AD 88 02	231	TEMY	LDA
CB9A: BD B6 02	232	TEMX	STB
CB93: B9 BB 06	233	OFFYL, Y	LDA
CB96: BD A1 02	234	OREGL	STB
CB99: B9 38 07	235	OFFYH, Y	LDA
CBAC: BD A2 02	236	OREGH	STB
CB9F: AD 83 02	237	YFL	LDA
CB82: AE B4 02	238	YFLH	LDX
CB85: 20 C2 CA	239	OFFDIV	JSR
CB88: 60	240	RTS	
CB99:	242	*	*****
CB99:	243	*	
CB99:	244	*	
CB99:	245	*	TRIPLE READ OF BITPAD
CB99:	246	*	IF OFFSCALE THEN A 1 IS
CB99:	247	*	PUT IN THE HIGH NIBBLE OF
CB99:	248	*	TEM
CB99:	249	*	
CB99:	250	*	
CB99:	251	*	*****
CB99: 20 F4 CB	252	MREAD	JSR
CBBC: 90 12	253	BCC	SWCHK
CBBE: 2C 00 C0	254	BIT	KBD
CBC1: 30 03	255	BMI	OFFS1
CBC3: 4C B9 CB	256	JMP	MREAD
CBC6: 48	257	OFFS1	PHA
CBC7: AD B0 02	258	LDA	TEM
CBCA: 09 08	259	DRA	#\$08
CBCC: BD B0 02	260	STA	TEM
CBCF: 68	261	PLA	
CBD0: 48	262	SWCHK	PHA
CBD1: 4D B0 02	263	EOR	TEM
CBD4: 6A	264	ROR	A
CBD5: 90 05	265	BCC	NOSWITCH
CBD7: A9 50	266	LDA	#\$50
CBD9: 20 A1 CC	267	JSR	GWAIT
CBDG: 68	268	NOSWITCH	PLA
CBDG: 6A	269	ROR	A
CBDE: AD B0 02	270	LDA	TEM
CBE1: 2A	271	ROL	A
CBE2: 29 13	272	AND	#\$13
CBE4: B0 B0 02	273	STA	TEM
CBE7: 4E B4 02	274	LSR	YFL
CBEA: 6E B3 02	275	ROR	YFL
CBED: 4E B2 02	276	LSR	XFL
CBF0: 6E B1 02	277	ROR	XFL

CBF3:60 278 RTS
 CBF4:AD F8 07 279 M1READ LDA MSLOT
 CBF7:0A 280 ASL A
 CBF8:0A 281 ASL A
 CBF9:0A 282 ASL A
 CBFA:0A 283 ASL A
 CBFB:AA 284 TAX
 CBFC:40 03 285 LDY #\$3
 CBFE:D0 0C 286 BNE C4SKP
 CCO0: 287 ORG \$400+OTHROM
 CCO1:80 288 CLC
 CCO2:38 289 DFB \$B0
 CCO3:88 290 SEC
 CCO4:08 291 CLV
 CCO5:78 292 PHP
 CCO6:2C FF CF 293 SEI
 CCO9:20 00 C8 294 BIT ROMSW
 CCO9:20 00 C8 295 JSR OTHROM
 CCOE:A9 00 296 C4SKP LDA #\$0
 CCOE:99 81 02 297 ZDLOOP STA XFLL,Y
 CC11:88 298 DEY
 CC12:10 FA 299 BPL ZDLOOP
 CC14:A0 06 300 LDY #\$6
 CC16:8C 78 06 301 STY HNDX
 CC19:20 83 CC 302 RDLOOP JSR RESLP
 CC1C:BD 81 C0 303 LDA DEVO,X
 CC1F:20 42 CC 304 JSR READTAB
 CC22:80 1D 305 BCS OFFSC
 CC24:A0 02 306 LDY #\$2
 CC26:20 8E CC 307 JSR AMOVE
 CC29:20 83 CC 308 JSR RESLP
 CC2C:BD 80 C0 309 LDA DEV1,X
 CC2F:20 42 CC 310 JSR READTAB
 CC32:80 0D 311 BCS OFFSC
 CC34:48 312 PHP
 CC35:A0 00 313 LDY #\$0
 CC37:20 8E CC 314 JSR AMOVE
 CC3A:68 315 PLA
 CC3B:CE 78 06 316 DEC HNDX
 CC3E:D0 D9 317 BNE RDLOOP
 CC40:18 318 CLC
 CC41:60 319 OFFSC RTS
 CC42:A9 12 320 READTAB LDA #\$12
 CC44:E9 01 321 ALLOOP SBC #\$1
 CC46:D0 FC 322 BNE ALLOOP
 CC48:BD 82 C0 323 LDA DEV3,X
 CC4B:0A 324 ASL A
 CC4C:0A 325 ASL A
 CC4D:0A 326 ASL A
 CC4E:0A 327 ASL A
 CC4F:49 70 328 EOR #\$70
 CC51:29 F0 329 AND #\$FO
 CC53:BD 87 02 330 STA TEMYL
 CC56:BD 83 C0 331 LDA DEV2,X
 CC59:BD 88 02 332 STA TEMY
 CC5C:A9 60 333 LDA #\$60
 CC5E:E9 01 334 ALLOOP SBC #\$1
 CC60:D0 FC 335 BNE ALLOOP
 CC62:A0 04 336 LDY #\$4
 CC64:4E 88 02 337 BDLOOP LSR TEMYL
 CC67:6E 87 02 338 ROR TEMYL
 CC6A:88 339 DEY
 CC6B:D0 F7 340 BNE DLLOOP
 CC6D:AD 88 02 341 LDA TEMY
 CC70:D0 07 342 BNE ATST
 CC72:A9 60 343 LDA #\$60
 CC74:CD 87 02 344 CMP TEMYL
 CC77:90 02 345 BCC BTST
 CC79:C9 0A 346 ATST CMP #\$0A
 CC7B:BD 82 C0 347 BTST LDA DEV3,X
 CC7E:A9 01 348 EOR #\$1
 CC80:29 01 349 AND #\$1
 CC82:60 350 RTS

*C400 SPACE ENTRY

CC83:AD FB 07 351 RESLP LDA MSLOT
 CC84:48 352 PHA
 CC87:A9 03 353 LDA#\$3
 CC89:48 354 PHA
 CC8A:2C 41 CC 355 BIT OFFSC
 CC8D:60 356 RTS
 CC8E:AD 87 02 357 AMOVE LDA TEMYL
 CC91:79 B1 02 358 ADC XFLL,Y
 CC94:99 B1 02 359 STA XFLL,Y
 CC97:AD 88 02 360 LDA TEMY
 CC9A:79 B2 02 361 ADC XFLL,Y
 CC9D:99 B2 02 362 STA XFLL,Y
 CCAD:60 363 RTS
 CCA1:28 364 GWAIT SEC
 CCA2:48 365 G2WAIT PHA
 CCA3:E9 01 366 G3WAIT SBC #\$1
 CCA5:D0 FC 367 BNE G3WAIT
 CCA7:68 368 PLA
 CCA8:E9 01 369 SBC #\$1
 CCA9:D0 F6 370 BNE G2WAIT
 CCAC:60 371 RTS
 CCAD: 373 *****
 CCAD: 374 *
 CCAD: 375 * CHARACTER ENTRY SEQUENCE
 CCAD: 376 *
 CCAD: 377 * CHARACTER STRING ANALYSIS
 CCAD: 378 * S... NNNNN SET SCALE FACTOR TO NNNNN
 CCAD: 379 * X... NNNNN SET X OFFSET TO NNNNN
 CCAD: 380 * Y... NNNNN SET Y OFFSET TO NNNNN
 CCAD: 381 * M... (1 OR 2) SET MIXED HIRES MODE
 CCAD: 382 * H... (1 OR 2) SET HIRES MODE
 CCAD: 383 * Q... (1 OR 2) SET MIXED LORES MODE
 CCAD: 384 * L... (1 OR 2) SET LORES MODE
 CCAD: 385 * T... (1 OR 2) SET TEXT MODE
 CCAD: 386 * N... SUPPRESS ALL PRINTING
 CCAD: 387 * D... RESTORE DEFAULT PARAMETERS
 CCAD: 388 * C... NO CURSOR
 CCAD: 389 * F... DATA RETURNED UNSCALED
 CCAD: 390 * R... DATA RETURNED SCALED
 CCAD: 391 * P... STREAM MODE ON
 CCAD: 392 * G... STREAM MODE OFF(DEFAULT)
 CCAD: 393 * A... OFFSET AFTER SCALE
 CCAD: 394 * B... OFFSET BEFORE SCALE(DEFAULT)
 CCAD: 395 *
 CCAD: 396 *****
 CCAD: 28 397 SYNTAX PLP TEST IF THIS IS THE FIRST CHAR
 CCAE:80 0C 398 BCS SYNT1 ; IF NOT THEN SKIP SET-UP
 CCBO:A9 00 399 LDA #\$0
 CCB2:BD 98 02 400 STA INX ; SET INDEX BUFFER TO ZERO
 CCB5:BD 99 02 401 STA NFLAG ; CLEAR BUFFER STATUS
 CCB8:A9 02 402 LDA #\$2 ; MOVE ENTRY POINTER
 CCBA:85 36 403 STA COUTL
 CCBC:AD FB 06 404 SYNT1 LDA CHAR ; LOAD THE ENTRY CHAR
 CCBF:C9 AD 405 CMP #\$AD ; TEST FOR MINUS
 CCC1:D0 03 406 BNE SYNT2 ; BRANCH IF NOT
 CCC3:BD A0 02 407 STA MIFLAG ; SET TO NOT ZERO
 CCC6:C9 A0 408 SYNT2 CMP #\$AO ; TEST FOR SPACE
 CCCB:F0 29 409 DEQ EXIT4 ; LEAVE IF SPACE
 CCCA:C9 AC 410 CMP #\$AC ; TEST FOR COMMA
 CCCC:F0 28 411 BEQ PROC1 ; IF SO - ANALYSE STRING
 CCCF:C9 BD 412 CMP #\$BD ; TEST FOR CR
 CCDD:F0 26 413 DEQ PROC2 ; IF SO - ANALYSE STRING
 CCDD:0B 414 PHP ; SAVE STATUS
 CCDD:AE 98 02 415 LDX INX ; BUFFER INDEX TO X-REG
 CCDD:F0 0C 416 BEQ CHOUT ; ASSUME ON ALPHA CHAR IF FIRST
 CCDB:49 B0 417 EOR #\$BO ; TEST FOR NUMERIC
 CCDA:C9 04 418 CMP #\$A
 CCDC:90 06 419 BCC CHOUT ; IF SO THEN BRANCH
 CCDE:E9 01 420 CPX #\$1 ; ITS OK TO GET HERE ONLY
 CCDE:F0 10 421 BEQ EXIT3 ; IF ONE ALPHA HAS BEEN STORED
 CCE2:D0 6C 422 BNE ERR1
 CCE4:E9 06 423 CHOUT CPX #\$6 ; ITS AN ERROR IF THERE ARE
 CCE6:B0 6B 424 BCS ERR1 ; ALREADY 5 CHARS IN THE BUFFER

CCEB: AD F8 06	425	LDA	CHAR	
CCEB: 9D 90 02	426	STA	INA, X	/ SAVE THE CHAR IN THE BUFFER
CCEE: E8	427	INX		/ UPDATE THE BUFFER INDEX
CCCF: BE 98 02	428	STX	INX	
CCF2: 28	429	EXIT3	PLP	/ RESTORE STATUS
CCF3: 4C 86 C8	430	EXIT4	JMP	/ GOTO EXIT
CCF6: 38	431	PROC1	SEC	/ SET CARRY IF A COMMA
CCF7: 90	432	DFB	#90	/ BCS ALWAYS
CCFB: 18	433	PROC2	CLC	/ CLEAR CARRY IF A CR
CCF9: 08	434	PHP		/ SAVE THE CARRY
CCFA: AD 90 02	435	LDA	INA	
CCFD: 4C 0C C0	436	JMP	C55KP	
CD00:	437	ORG	\$500+0THROM	
CD00: 18	438	CLC		; \$C500 SPACE ENTRY
CD01: B0	439	DFB	#80	
CD02: 38	440	SEC		
CD03: B8	441	CLV		
CD04: 08	442	PHP		
CD05: 78	443	SEI		
CD06: 2C FF CF	444	BIT	ROMSW	
CD09: 20 00 C8	445	JSR	0THROM	
CD0C: A0 12	446	C55KP	LDY	#\$12
CD0E: B8	447	ALPH1	DEY	
CD0F: F0 3F	448	BEG	ERR1	
CD11: D9 5C CE	449	CMP	TABL-1, Y	
CD14: D0 F8	450	BNE	ALPH1	
CD16: A9 CD	451	LDA	WROUTIN	
CD18: 48	452	PHA		
CD19: B9 6D CE	453	LDA	ADR-1, Y	
CD1C: 48	454	PHA		
CD1D: B9 7E CE	455	LDA	PARAM-1, Y	
CD20: AC 98 02	456	LDY	INX	
CD23: 60	457	RTS		; JUMP TO ROUTINE WITH RTS
CD24: C0 01	458	ROUTIN	CPY	#\$1
CD26: D0 28	459	BNE	ERR1	
CD28: C9 CE	460	CMP	#\$CE	
CD2A: D0 05	461	BNE	ROUT1	
CD2C: BD 99 02	462	STA	NFLAG	
CD2F: F0 18	463	BEG	EXITS5	
CD31: AE F8 07	464	ROUT1	LDX	MSLOT
CD34: C9 C4	465	CMP	#\$C4	
CD36: D0 05	466	BNE	ROUT2	
CD38: 20 90 CE	467	JSR	DEFAULT	
CD3B: D0 0F	468	BNE	EXITS5	
CD3D: 1E B8 03	469	ROUT2	ASL	PAGE, X
CD40: 48	470	PHA		
CD41: 68	471	PLA		
CD42: F0 04	472	DEG	NDCRS	
CD44: 0A	473	ASL	A	
CD45: BD B8 03	474	LDA	PAGE, X	
CD48: 6A	475	NDCRS	ROR	A
CD49: 9D B8 03	476	STA	PAGE, X	
CD4C: 4C 0C CE	477	EXITS5	JMP	EXIT1
CD4F: 68	478	ERR3	PLA	
CD50: 20 93 FE	479	ERR1	JSR	SETVID
CD53: A0 19	480	EMSG	LDY	#\$19
CD55: B9 43 CE	481	EMSG1	LDA	STRIN-1, Y
CD58: 20 ED FD	482	JSR	COUT	
CD5B: B8	483	DEY		
CD5C: D0 F7	484	BNE	EMSG1	
CD5E: AD 91 C0	485	LDA	STEXT	
CD61: AD 54 C0	486	LDA	SPA01	
CD64: 4C 0C CE	487	JMP	EXIT1	
CD67: C0 02	488	ARCUT	CPY	#\$2
CD69: D0 E5	489	BNE	ERR1	
CD6B: 48	490	PHA		
CD6C: AD 91 02	491	LDA	INA+1	
CD6F: 49 B0	492	EDR	#\$B0	
CD71: F0 DC	493	BEG	ERR3	
CD73: C9 03	494	CMP	#\$3	
CD75: B0 D8	495	BCS	ERR3	
CD77: 6A	496	ROR	A	
CD78: 68	497	PLA		

CD79: B0 01	498	BGS	NOROL	
CD7B: 2A	499	ROL	A	
CD7C: AE F8 07	500	NOROL	LDX	MSLOT
CD7F: 1E B8 03	501	ASL	PAGE, X	
CDB2: 6A	502	ROR	A	
CDB3: 20 BE CE	503	JSR	STMODE	
CDB6: 4C 0C CE	504	JMP	EXIT1	
CDB9: B5 2B	505	BROUT	STA	HBASH
CDB9: A9 B8	506	LDA	#\$BB	
CDBD: B5 2A	507	STA	HBASL	
CDBF: B8	508	DEY		
CD90: B9 90 02	509	DIGLP	LDA	INA, Y
CD93: 49 B0	510	EDR	#\$B0	
CD95: C9 0A	511	CMP	#\$A	
CD97: B0 B7	512	BCS	ERR1	
CD99: 99 90 02	513	STA	INA, Y	
CD9C: B8	514	DEY		
CD9D: D0 F1	515	BNE	DIGLP	
CD9F: A2 00	516	LDX	#\$0	
CDA1: C8	517	INY		
CDA2: B0 7B 05	518	DIG2	STA	TEMPL
CDA3: B8 F8 03	519	STX	TEMPH	
CDA8: C8	520	INY		
CDA9: CC 98 02	521	CPY	INX	
CDAE: F0 1D	522	DEG	DIG3	
CDAE: A9 0A	523	LDA	#\$A	
CDB0: B0 A5 02	524	STA	COUNT	
CDB3: A2 00	525	LDX	#\$0	
CDB5: B9 90 02	526	LDA	INA, Y	
CDBB: B0 7B 05	527	DIG1	ADC	TEMPL
CDBC: 4B	528	PHA		
CDBC: B8	529	TXA		
CDBD: 60 F8 05	530	ADC	TEMPH	
CDC0: 30 B8	531	BMI	ERR1	
CDC2: AA	532	TAX		
CDC3: 6B	533	PLA		
CDC4: CE A5 02	534	DEC	COUNT	
CDC7: D0 EF	535	BNE	DIG1	
CDC9: F0 D7	536	DEG	DIG2	
CDCB: A4 2B	537	DIG3	LDY	HBASH
CDCD: C0 04	538	CPY	#\$4	
CDCF: F0 08	539	BEG	DIG4	
CDD1: AC A0 02	540	LDY	MIFLAG	
CDD4: F0 03	541	BEGDIG4		
CDD6: 20 5B C8	542	JSR	TWOCOM	
CDD9: AC F8 07	543	DIG4	LDY	MSLOT
CDDC: 91 2A	544	STA	(HBASL), Y	
CDEE: A9 3B	545	LDA	#\$3B	
CDEO: 85 2A	546	STA	HBASL	
CDE2: E6 2B	547	INC	HBASH	
CDE4: B8	548	TXA		
CDE5: 91 2A	549	STA	(HBASL), Y	
CDE7: 4C 0C CE	550	JMP	EXIT1	
CDEA: AC F8 07	551	CROUT	LDY	MSLOT
CDED: 4B	552	PHA		
CDEE: 6A	553	ROR	A	
CDEF: 6B	554	PLA		
CDF0: 90 05	555	BCC	CROUT2	
CDF2: 39 3B 04	556	AND	MPAGE, Y	
CDF5: B0 03	557	BCS	CROUT3	
CDF7: 19 3B 04	558	ORA	MPAGE, Y	
CDF8: 99 3B 04	559	STA	MPAGE, Y	
CDFD: 4C 0C CE	560	JMP	EXIT1	
CED0: 1B	561	ORG	\$600+0THROM	
CE00: 1B	562	CLC		; \$C600 SPACE ENTRY
CE01: B0	563	DFB	#\$B0	
CE02: 3B	564	SEC		
CE03: B8	565	CLV		
CE04: 08	566	PHP		
CE05: 7B	567	SEI		
CE06: 2C FF CF	568	BIT	ROMSW	
CE09: 20 00 C8	569	JSR	0THROM	
CE0C: A9 00	570	EXIT1	LDA	#\$0

CE0E:8D A0 02 571	STA	MIFLAG
CE11:8D 98 02 572	STA	INX
CE14:AC F8 07 573	LDY	MSLOT
CE17:B9 38 04 574	LDA	MPAGE, Y
CE1A:29 C0 575	AND	#\$CO
CE1C:99 38 04 576	STA	MPAGE, Y
CE1F:B9 B8 03 577	LDA	PAGE, Y
CE22:29 3F 578	AND	#\$3F
CE24:49 25 579	EOR	#\$25
CE26:19 38 04 580	ORA	MPAGE, Y
CE29:99 38 04 581	STA	MPAGE, Y
CE2C:29 582	PLP	
CE2D:80 12 583	BCS	EXIT2
CE2F:AD 99 02 584	LDA	NFLAG
CE32:F0 0A 585	BEG	PRINT
CE34:A9 58 586	LDA	#IORTS
CE36:B5 36 587	STA	COUTL
CE38:A9 FF 588	LDA	#IORTS/256
CE3A:B5 37 589	STA	COUTH
CE3C:D0 03 590	ONE	EXIT2
CE3E:20 93 FE 591	JSR	SETVID
CE41:4C 86 CB 592	JMP	EXIT
CE44:8D 593	STRIN	DFB \$8D
CE45:D2 CF D2 594	ASC	"RORRE
XATNYS TELBAT ***"		
CE48:D2 C5 A0		
CE4B:D8 C1 D4		
CE4E:CE D9 D3		
CE51:A0 D4 C5		
CE54:CC C2 C1		
CE57:D4 A0 AA		
CESA:AA AA B7		
CE5D:D0 D1 C1 595	TABL	ASC "PGABSXVTMHLGCFRND"
CE60:C2 D3 D8		
CE63:D9 D4 CD		
CE66:C8 CC C7		
CE69:C3 C6 D2		
CE6C:CE C4		
CE6E:E9 596	ADR	DFB CROUT-1
CE6F:E9 597	DFB	CROUT-1
CE70:E9 598	DFB	CROUT-1
CE71:E9 599	DFB	CROUT-1
CE72:B8 600	DFB	BROUT-1
CE73:B8 601	DFB	BROUT-1
CE74:B8 602	DFB	BROUT-1
CE75:B8 603	DFB	AROUT-1
CE76:B8 604	DFB	AROUT-1
CE77:B8 605	DFB	AROUT-1
CE78:B8 606	DFB	AROUT-1
CE79:B8 607	DFB	AROUT-1
CE7A:23 608	DFB	ROUTIN-1
CE7B:23 609	DFB	ROUTIN-1
CE7C:23 610	DFB	ROUTIN-1
CE7D:23 611	DFB	ROUTIN-1
CE7E:23 612	DFB	ROUTIN-1
CE7F:80 613	PARAM	DFB \$80
CE80:7F 614	DFB	\$7F
CE81:40 615	DFB	\$40
CE82:8F 616	DFB	\$BF
CE83:04 617	DFB	\$04
CE84:05 618	DFB	\$05
CE85:06 619	DFB	\$06
CE86:08 620	DFB	\$08
CE87:48 621	DFB	\$48
CE88:40 622	DFB	\$40
CE89:02 623	DFB	\$02
CE8A:0A 624	DFB	\$0A
CE8B:00 625	DFB	\$0
CE8C:01 626	DFB	\$01
CE8D:80 627	DFB	\$80
CE8E:CE 628	DFB	\$CE
CE8F:C4 629	DFB	\$C4

CE90:631	*****	*****	*****	*****
CE90:632	*****	*****	*****	*****
CE90:633	*			
CE90:634	* DEFAULT			
CE90:635	*			
CE90:636	* SCALE = 16			
CE90:637	* X OFFSET = 1536			
CE90:638	* Y OFFSET = 1536			
CE90:639	* HIRES PAGE 2			
CE90:640	* PRINT ON			
CE90:641	* CURSOR ON			
CE90:642	* STREAM MODE OFF			
CE90:643	* DATA RETURNED UNSCALED			
CE90:644	* OFFSET BEFORE SCALE			
CE90:645	*			
CE90:646	*****	*****	*****	*****
CE90:647	*****	*****	*****	*****
CE90:A9 10 648	DEFAULT	LDA	#\$10	
CE92:9D B8 04 649	STA	SCALL, X		
CE95:A9 00 650	LDA	#\$0		
CE97:9D B8 05 651	STA	SCALH, X		:SET SCALE FOR 16
CE9A:9D B8 05 652	STA	OFFXL, X		
CE9D:9D B8 06 653	STA	OFFYL, X		:SET X OFFSET TO 1536
CEA0:8D B8 02 654	STA	INX		
CEA3:A9 03 655	LDA	#\$3		
CEA5:BD B0 02 656	STA	TEM		
CEAB:A9 06 657	LDA	#\$6		:SET Y OFFSET TO 1536
CEAA:9D B8 06 658	STA	OFFXH, X		
CEAD:9D B8 07 659	STA	OFFYH, X		
CEB0:20 10 C0 660	BIT	KBDSTRB		
CEB3:20 F4 CB 661	JSR	MIREAD		
CEB6:20 D0 CB 662	JSR	SWCHK		
CEB9:AE FB 07 663	LDX	MSLOT		
CEBC:A9 40 664	LDA	#\$40		:SET PAGE FOR HIRES P2
CEBE:AB 665	STMODE	TAY		
CEBF:8D B2 C0 666	STA	SNMIX		
CEC2:BD B7 C0 667	STA	SHIRES		
CEC5:BD B1 C0 668	STA	STEXT		
CEC8:BD B4 C0 669	STA	SPAG1		
CECB:29 0C 670	AND	#\$0C		
CECD:F0 03 671	BEG	DEF1		
CECF:BD B3 C0 672	STA	SMIX		
CED2:98 673	DEF1	TYA		
CED3:29 43 674	AND	#\$43		
CED5:F0 03 675	BEG	DEF2		
CED7:AD 50 C0 676	LDA	SGR		
CEDA:98 677	DEF2	TYA		
CEDB:29 4A 678	AND	#\$4A		
CEDD:F0 03 679	BEG	DEF3		
CEDF:AD 55 C0 680	LDA	SPAG2		
CEE2:98 681	DEF3	TYA		
CEE3:29 03 682	AND	#\$03		
CEE5:F0 03 683	BEG	DEF4		
CEE7:BD B6 C0 684	STA	SLDRES		
CEE8:98 685	DEF4	TYA		
CEE9:9D B8 03 686	STA	PAGE, X		:AND FOR FULL SCALE OUTPUT
CEE9:29 3F 687	AND	#\$3F		
CEFO:49 25 688	EOR	#\$25		
CEF2:9D B8 04 689	STA	MPAGE, X		
CEF5:60 690	RTS			
CEF6:691	*****	*****	*****	*****
CEF6:692	*			
CEF6:693	* JUMP TABLE			
CEF6:694	*			
CEF6:695	*****	*****	*****	*****
CEF6:696	ORG	\$6F6+0THROM		
CEF6:4C 4C CA 697	JWINCHK	JMP	WINCHK	
CEF9:4C B9 CB 698	JMREAD	JMP	MREAD	
CEFC:4C 70 CB 699	JSCALE	JMP	SCALE	
*** SUCCESSFUL ASSEMBLY: NO ERRORS				

CC5E ATLOOP	CE6E ADR	CD0E ALPH1	CC8E AMOVE
CB09 AOTHROM	CD67 AROUT	C9B3 A	CC44 ALOOP
CB00 ASC1EX	CBC6 ASCEX	CA6A ASCON	CA89 ASKIP
CC79 ATBT	C9C4 B	C9F0 BASCLC	28 BASL
CD89 BROUT	CAAO BSKIP	CC7B BTST	02A5 C1HAR
C90C C1SKP	CA0C C2SKP	CC0C C4SKP	CD0C C5SKP
7C918 CALLCURS	CCE4 CHOUT	24 CH	06F8 CHAR
CA39 GLOOP	02A5 COUNT	FDE0 COUT	37 COUTH
36 COUTL	CDF7 CROUT2	C0FA CROUT3	C8F0 CURSOUT
C9AD C	CDEA CROUT	CAB2 CSKIP	C9D8 CTRCHK
C995 CTRLLOOP	CED2 DEF1	CEA4 DEF2	CEE2 DEF3
CEEA DEF4	CE90 DEFAULT	C081 DEVO	C080 DEV1
CD83 DEV2	C082 DEV3	7CB2D DEXIT	CDB8 DIG1
CD42 DIV2	CDCB DIV3	CDD9 DIV4	CD90 DIGLP
02A8 DIVH	CAD1 DIVIDE	02A3 DIVL	CC64 DLOOP
CABD DSHPKP	CB57 E1EXIT	C868 E1PNT	C83F EEXIT
CD55 EM6G1	7CD53 EM6G	C86D END	C83F EPOINT
CD50 ERR1	CD4F ERR3	CE0C EXIT1	CCF2 EXIT3
CD4C EXIT5	C886 EXIT	CE41 EXIT2	CCF3 EXIT4
CB2C FEXIT	CA86 FLOOP	CCA2 Q2WAIT	CCA3 Q3WAIT
CB90 GLOOP	C951 GR	CCA1 QWAIT	28 HBASH
2A HBASL	C971 HIRES	CA9D HLOOP	0678 HNDX
CA2B HPOSN1	CA2C HPOSN2	0200 IN0	0201 IN1
0202 IN2	0203 IN3	0290 INA	C86F INEX1
0883 INEX2	CBBC INEXIT	7C900 INPUTXY	0298 INX
FF58 IORTS	CAAF JLOOP	7CEF9 JMREAD	7CEFC JSSCALE
7CEF6 JWINCHK	C000 KBD	C010 KBDSTRB	CABA KLOOP
CE9E LOCLC	C980 LOOP	CAD7 LOOP1	CAF3 LOOP2
CB0C _LOOP3	7CB1F LOOP4	C962 LOR1	7C95A LORES
CBF4 MIREAD	02A0 MIFLAG	7C930 MIX	0438 MPAGE
CB89 MREAD	07F8 MSLOT	0299 NFLAG	CA67 NO
CD48 NOCRS	CB34 NOPR	CD7C NOROL	CBDC NOSWITCH
CAC2 OFFDIV	CBC6 OFFS1	CC41 OFFSC	CB64 OFFSET
0638 OFFXH	05B8 OFFXL	0738 OFFYH	06B8 OFFYL
02A2 OREGH	02A1 OREOL	CB00 OTHROM	C94E OUT1
C99E OUTSIDE	C9D0 OUT	03B8 PAGE	CE7F PARAM
7C902 POINT	CA80 POSIT	CAF2 POS	C82D PRCHK
CE3E PRINT	CCF6 PROC1	CCF8 PROC2	CC19 RDLDOP
CC42 READTAB	0288 REGH	0287 REGL	CCB3 REBUP
CFFF ROMSW	CD31 ROUT1	CD3D ROUT2	CD24 ROUTIN
7CAC1 RTN	0298 SAVSLOT	C870 SCALE	0538 SCALH
04B8 SCALL	FE93 SETVID	C050 SGR	C057 SHIRES
C056 SLORES	C053 SMIX	C052 SNMIX	C054 SPAQ1
C055 SPAQ2	C051 STEXT	CE8E STMODE	CE44 STRIN
CB00 SWCHK	CCBC SYNT1	CCC6 SYNT2	CCAD SYNTAX
CE5D TABL	05F8 TEMPH	0578 TEMPL	02B5 TEMXL
0286 TEMX	0287 TEMYL	0288 TEMY	02B0 TEM
C936 TEXT	CB58 TWOCOM	CA4C WINCHK	02B2 XFLH
02B1 XFLL	CASF YES	0284 YFLH	02B3 YFLL
CC0E ZDLOOP			
24 CH	28 BASL	2A HBASL	28 HBASH
36 COUTL	37 COUTH	0200 IN0	0201 IN1
0202 IN2	0203 IN3	0280 TEM	02B1 XFLL
02B2 XFLH	02B3 YFLH	0284 YFLH	02B5 TEMXL
02B6 TEMX	02B7 TEMYL	0287 REGL	02B8 TEMY
02B9 REGH	0290 INA	0298 INX	0299 NFLAG
02B9 SAVSLOT	02A0 MIFLAG	02A1 OREOL	02A2 OREGH
02A3 DIVL	02A4 DIVH	02A5 C1HAR	02A5 COUNT
03B8 PAGE	0438 MPAGE	04B8 SCALL	0538 SCALH
0578 TEMPL	05B8 OFFXL	05F8 TEMPH	0638 OFFXH
0678 HNDX	06B8 OFFYL	06F8 CHAR	0728 OFFYH
07F8 MSLOT	C000 KBD	C010 KBDSTRB	C050 SGR
C051 STEXT	C052 SNMIX	C053 SMIX	C054 SPAQ1
C055 SPAQ2	C056 SLORES	C057 SHIRES	C080 DEV1
CB81 DEVO	C082 DEV3	C083 DEV2	CB00 OTHROM
CB09 AOTHROM	C82D PRCHK	C834 NOPR	C83F EPOINT
C868 E1PNT	C86D END	C86F INEX1	CB83 INEX2
C886 EXIT	CBBC INEXIT	C890 GLOOP	CBC6 ASCEX
CB00 ASC1EX	CBF0 CURSOUT	7C900 INPUTXY	7C902 POINT
C90C C1SKP	7C918 CALLCURS	7C930 MIX	C936 TEXT
C94E OUT1	C951 GR	7C95A LORES	C962 LOR1
C971 HIRES	C980 LOOP	C985 CTRLLOOP	C99E OUTSIDE
C9AD C	C983 A	C9C4 B	C9D0 OUT

C9D8 CTRCHK	C9DE LOCLC	C9F0 BASCLC	CA0C C2SKP
CA2B HPOSN1	CA2C HPOSN2	CA39 CLOOP	CA4C WINCHK
CASF YES	CA67 NO	CA6A ASCON	CA80 POSIT
CAB6 FLOOR	CA89 ASKIP	CA9D HLOOP	CAA0 BSKIP
CAAF JLOOP	CAB2 CSKIP	CABA KLOOP	CABD DSKIP
?CAC1 RTN	CA12 OFFDIV	CAD1 DIVIDE	CAD7 LOOP1
CAF2 POS	CAF3 LOOP2	CB0C LOOP3	?CB1F LOOP4
CB2C FEXIT	?CB2D DEXIT	CB3F EEXIT	CB57 E1EXIT
CB5B TWOCOM	CB64 OFFSET	CB70 SCALE	CBB9 MREAD
CBC6 OFFS1	CB00 SWCHK	CBDC NOSWITCH	CBF4 M1READ
CC0C C4SKP	CC0E ZDLOOP	CC19 RDLOOP	CC41 OFSC
CC42 READTAB	CC44 ALOOP	CC5E A1LOOP	CC64 DLOOP
CC79 ATST	CC7B BTST	CCB3 RESLP	CC8E AMOVE
CCA1 GWAIT	CCA2 Q2WAIT	CCA3 Q3WAIT	CCAD SYNTAX
CCBC SYNT1	CCC6 SYNT2	CCE4 CHOUT	CCF2 EXIT3
CCF3 EXIT4	CCF6 PROC1	CCFB PROC2	CDOC C5SKP
CD0E ALPH1	CD24 ROUTIN	CD31 ROUT1	CD3D ROUT2
CD48 NOCRS	CD4C EXIT5	CD4F ERR3	CD50 ERR1
?CD53 EMSG	CD95 EMS01	CD67 AROUT	CD7C NOROL
CD89 BROUT	CD90 DIGLP	CDA2 DIG2	CD88 DIG1
CDCB DIG3	CDD9 DIQ4	CDEA CROUT	CDF7 CROUT2
CDFA CROUT3	CE0C EXIT1	CE3E PRINT	CE41 EXIT2
CE44 STRIN	CE5D TABL	CE6E ADR	CE7F PARAM
CE90 DEFAULT	CEBE STMODE	CED2 DEF1	CEDA DEF2
CEE2 DEF3	CEEA DEF4	?CEF6 JWINCHK	?CEF9 JMREAD
?CEFC JSCALE	CFFF ROMSW	FD0D COUT	FE93 SETVID
FF58 IDRTS			

QUICK-DRAW

```

0000      1      PAGE
0000      2 ****
0000      3 *
0000      4 * COPYRIGHT 1979
0000      5 *
0000      6 * APPLE COMPUTER INC.
0000      7 *
0000      8 * CUPERTINO CALIFORNIA
0000      9 *
0000     10 * ALL RIGHTS RESERVED
0000     11 *
0000     12 ****
0000     13 * WRITTEN JAN 1979
0000     14 * BY JOHN A.
0000     15 * APPLE COMPUTER
0000     16 * SYSTEMS SOFTWARE
0000     17 ****
0000     18 * WINDOCK CORRECTED APR 25, 1979
0000     19 * BY JOHN A.
0000     20      PAGE
0000     21 ****
0000     22 *
0000     23 * BITPAD TO APPLESOFTII *
0000     24 * INTERFACE ROUTINE *
0000     25 * THIS ROUTINE MAKES *
0000     26 * IT POSSIBLE TO CALL *
0000     27 * THE BITPAD AT HIGH *
0000     28 * SPEED AND FETCH THE *
0000     29 * POINTS DRAWN DIRECTLY*
0000     30 * INTO APPLESOFT DATA *
0000     31 * ARRAYS AT MAXIMUM *
0000     32 * SPEED. ARRAYS MUST BE *
0000     33 * DIMENSIONED BEFORE *
0000     34 * CALLING THIS ROUTINE *
0000     35 * THEY ARE XX AND YX *
0000     36 * IN ADDITION NX AND DX *
0000     37 * MUST ALSO BE ALLOCATED*
0000     38 ****
0000     39      ORG $C00
0000     40      OBJ $2000
0000     41 ****
0000     42 *PAGE ZERO USAGE *
0000     43 VARPNT      EQU $83
0000     44 VARNAM      EQU $81
0000     45 LOWTR       EQU $98
0000     46 TXTTAB       EQU $67
0000     47 VARTAB       EQU $69
0000     48 ARYTAB       EQU $68
0000     49 STREND       EQU $6D
0000     50 ARYPNT       EQU $94
0000     51 *FAC EQU $9D TO A3
0000     52 DELTA        EQU $9D
0000     53 INDX        EQU $9E
0000     54 NADRS       EQU $A0
0000     55 MAXN        EQU $A2
0000     56 *ARG EQU $A5 TO AB
0000     57 TICFLG       EQU $A5
0000     58 XVPTR        EQU $A6
0000     59 YVPTR        EQU $A8
0000     60 TMAXY        EQU $AA      TEMP MAX Y COORD LIMIT
0000     61 * TMAXY = MIN(MIXED MODE*160, MAXY)
0000     62 XOL          EQU $E0      PRIOR X-COORD SAVE
0000     63 XOH          EQU $E1      PRIOR X-COORD SAVE HI
0000     64 YO           EQU $E2      PRIOR Y-COORD SAVE
0000     65 ERRFLG       EQU $D8
0000     66 REMSTK       EQU $FB
0000     67 ERRNUM       EQU $DE
0000     68 ****
0000     69 *ENTRY POINTS USED *

```

```

0C00:      70 HILIN      EQU $F53A      : REAL HILIN ENTRY
0C00:      71 HPLOT      EQU $F457      : REAL HPLOT ENTRY
0C00:      72 CHRGOT     EQU $00B1      : NEXT PGH CHAR
0C00:      73 CRDO       EQU $DAFB
0C00:      75 OUTOST     EQU $DB5A
0C00:      76 OUTDO      EQU $DB5C
0C00:      77 ISLETC     EQU $E07D
0C00:      78 TYPERR     EQU $D42A
0C00:      79 HNDLERX    EQU $F2EF
0C00:      80 WAIT       EQU $FCAB      : MON A WAIT
0C00:      81 PAGE       PAGE
0C00:      82 ****
0C00:      83 * DEVICE ADDRESSES
0C00:      84 SPKR        EQU $C030      : TOGGLS APPLE SPKR
0C00:      85 STEXT        EQU $C051      : SET TEXT MODE
0C00:      86 SPAG1       EQU $C054
0C00:      87 ****
0C00:      88 * BITPAD INTERFACE EQU *
0C00:      89 ****
0C00:      90 TEM         EQU $280      : RETURN FLAG LOC
0C00:      91 * HI NIBBLE =1 FOR SCALED RESULTS
0C00:      92 * LO NIBBLE 0=OPEN DOWN
0C00:      93 * 1=PEN LIFT
0C00:      94 * 2 PEN FALL
0C00:      95 * 3=PEN UP
0C00:      96 XFLL        EQU $281      : X-COORD LO UNSCALED
0C00:      97 XFLH        EQU $282      : X-COORD HI UNSCALED
0C00:      98 YFLL        EQU $283      : Y-COORD LOW UNSCALED
0C00:      99 YFLH        EQU $284      : Y-COORD HI UNSCALED
0C00:     100 TEMXL      EQU $285      : X-COORD LO SCALED
0C00:     101 TEMX       EQU $286      : X-COORD HI SCALED
0C00:     102 TEMYL      EQU $287      : Y-COORD LO SCALED
0C00:     103 TEMY       EQU $288      : Y-COORD HI SCALED
0C00:     104 SSM1        EQU $29A      : LO INDIRECT ADRS
0C00:     105 SAVSLOT     EQU $29B
0C00:     106 RTNCD      EQU $28C      : +700 BITSOFT RTN CODE
0C00:     107 PAGE        EQU $388      : +CN : SCREEN MODE
0C00:     108 * HI BIT=1 MEANS SCALE DATA
0C00:     109 * 40 = HIRES PG2
0C00:     110 * 20 = HIRES PG1
0C00:     111 HIRES1     EQU $20
0C00:     112 * 08 = TEXT PG2
0C00:     113 * 04 = TEXT PG1
0C00:     114 * 02 = LORES PG2
0C00:     115 * 01 = LORES PG1
0C00:     116 * 42 = HIRES PG2 MIXED
0C00:     117 * 21 = HIRES PG1 MIXED
0C00:     118 * 0A = LORES PG2 MIXED
0C00:     119 * 05 = LORES PG1 MIXED
0C00:     120 MXYVALU    EQU 160      : FOR NORMAL APPLE
0C00:     121 MSLOT       EQU $7FB
0C00:     122 ****
0C00:     123 * BITPAD ENTRY POINTS *
0C00:     124 ****
0C00:     125 POINT       EQU $C102
0C00:     126 MREAD      EQU $CEF9
0C00:     127 WINCHK     EQU $CEF6
0C00:     128 SCALE       EQU $CEFC
0C00:     129 PAGE       PAGE
0C00:     130 ****
0C00:     131 JMP BITSOFT
0C00:     132 JMP FINDVAR      : FIND VARIABLE UTILITY ENTR
0C00:     133 JMP FINDARY      : FIND ARRAY UTILITY ENTRY
0C00:     134 ****
0C00:     135 VNAMTAB     EQU *
0C00:     136 DLTNAM      DFB $C4, $80      : DX
0C00:     137 NDXNAM      DFB $CE, $80      : NX
0C00:     138 XVNAM       DFB $DB, $80      : XX
0C00:     139 YVNAM       DFB $D9, $80      : YZ
0C00:     140 DNAME        EQU DLTNAM-VNAMTAB
0C00:     141 NNAME        EQU NDXNAM-VNAMTAB
0C00:     142 XNAME        EQU XVNAM-YVNAMTAB

```

OC11	143	VNAME	EGU YVNAME-VNAMTAB		
OC11	00	144	MINXL	DFB \$00	
OC12	00	145	MINX	DFB \$00	
OC13	18	146	MAXXL	DFB 24	
OC14	01	147	MAXX	DFB 1	, MAX X=256+24
OC15	00	148	MINY	DFB \$0	
OC16	00	149	MAXY	DFB 192	
OC17	03 4F 00	150	CTABLE	DFB \$03, \$4F, \$00	
OC1A	19 52 09	151		DFB \$19, \$52, \$09	
OC1D	07 48 04	152		DFB \$07, \$48, \$04	
OC20	20 41 00	153		DFB \$20, \$41, \$00	
OC23	10 40 05	154		DFB \$10, \$40, \$05	
OC26	20 43 0F	155		DFB \$20, \$43, \$0F	
OC29	0D 50 05	156		DFB \$0D, \$50, \$05	
OC2C	14 45 02	157		DFB \$14, \$45, \$02	
OC2F	37 79	158		DFB \$37, \$79	
OC31		159		PAGE	
OC31		160	*****		
OC31		161	*	BITPAD MESSAGES	
OC31	20 44 41				
OC34	50 54 49				
OC37	C2	162	BITMSG	DCI "	DAPTIB"
OC38	44 45 4C				
OC3B	54 41 20				
OC3E	53 49 5A				
OC41	C5	163	DLTASIZ	DCI "DELTA	SIZE"
OC42	44 45 4C				
OC45	54 41 20				
OC48	55 4E 44				
OC4B	45 46 49				
OC4E	4E 45 C4	164	DLTADEF	DCI "DELTA	UNDEFINED"
OC51	49 4E 44				
OC54	45 58 20				
OC57	55 4E 44				
OC5A	45 46 49				
OC5D	4E 45 C4	165	INDXDEF	DCI "INDEX	UNDEFINED"
OC60	41 52 52				
OC63	41 59 20				
OC66	44 49 4D				
OC69	45 4E 53				
OC6C	49 4F CE	166	ARYDEF	DCI "ARRAY	DIMENSION"
OC6F	42 41 44				
OC72	20 53 55				
OC75	42 53 43				
OC78	52 49 50				
OC7B	D4	167	NGTMAXN	DCI "BAN	SUBSCRIPT"
OC7C	4E 4F 54				
OC7F	20 49 4E				
OC82	20 53 43				
OC85	41 4C 45				
OC88	44 20 48				
OC8B	49 52 45				
OC8E	53 20 4D				
OC91	4F 44 C5	168	NOTHGR	DCI "NOT	IN SCALED HIRES MODE"
OC94		169	DLTASIZ	EGU DLTASIZ-BITMSG+40	: THIS BIASES RTN CDS BY
OC94		170	DLTADEF	EGU DLTADEF-BITMSG+40	: THIS BIASES RTN CDS BY
OC94		171	XINDXDEF	EGU INDXDEF-BITMSG+40	: THIS BIASES RTN CDS BY
OC94		172	XARYDEF	EGU ARYDEF-BITMSG+40	: THIS BIASES RTN CDS BY 4
OC94		173	XNRANGE	EGU NGTMAXN-BITMSG+40	: THIS BIASES RTN CDS BY
OC94		174	XNOTHGR	EGU NOTHGR-BITMSG+40	: THIS BIASES RTN CDS BY 4
OC94		175		PAGE	
OC94		176	*****		
OC94		177	*	BEGIN CODE	*
OC94		178	*****		
OC94		179	BITSOFT	EGU *	
OC94	20 95 0F	180		JSR USRNAMS	
OC97	A2 00	181	NONAMES	LDX #INNAME	:=0
OC99	20 E4 0E	182		JSR SETNAME	: SET UP VARNA
OC9C	20 58 0F	183		JSR FINDVAR	: GO FIND DELTA
OC9F	B0 33	184		BCS DLTAOK	: HE DID ALLOCATE IT!
OC41	A2 39	185		LDX #DLTADEF	
OC43	186	BITPERR		EGU *	
OC43	24 00	187		BITERRFLG	: IS ON ERR ON?

OC45	10 0B	188		BPL DOERR	: NO
OC47	86 DE	189		STX ERRNUM	
OC49	A6 FB	190		LDX REMSTK	: GET STACK PTR
OCAB	9A	191		TXS	: RESTORE STACK
OCAC	4C EF F2	192		JMP HNDLERX	
OCAF		193	DOERR	EGU *	
OCAF	AD 54 C0	194		LDA SPAG1	: BACK TO PAGE 1 TOO
OCB2	AD 51 C0	195		LDA STEXT	: BACK TO TEXT MODE!
OCB5	20 F3 DA	196		JSR CRDO	
OCBB	20 5A DB	197		JSR OUTGST	
OCBD	A0 06	198		LDY #6	
OCBO	B9 31 OC	199	BITPD	LDA BITMSG, Y	
OCBO	20 5C DB	200		JSR OUTDO	
OCCE	88	201		DEY	
OCCE	10 F7	202		BPL BITPD	
OCCE	BD 09 OC	203	BMSGP	LDA BITMSG-40, X	
OCCE	48	204		PHA	
OCCE	20 5C DB	205		JSR OUTDO	
OCCE	E8	206		INX	
OCCE	68	207		PLA	
OCCE	10 F5	208		BPL BMSGP	: LOOP FOR NXT
OCD1	4C 2A D4	209		JMP TYPERR	: ADD LINE # & QUIT
OCD4	A0 00	211		EGU *	
OCD6	B1 83	212		LDA (VARPN), Y	
OCDE	B4 A5	213		STY TICFLQ	: ASSUME TICK
OCDA	18	214		CLC	: ASSUME NOT NEGATIVE
OCDB	10 0B	215		BPL CHKDVALU	
OCDD	38	216		SEC	: SAY NEGATIVE
OCDE	B5 A5	217		STA TICFLQ	: SET NOTICK
OCDE	49 FF	218		EDR #FF	: IS DX < 256
OCDE	F0 04	219		BEQ CHKDVALU	: IT'S OK
OCDE	A2 2F	220	DSIZERR	LDX #XDLTASIZ	
OCDE	BD BB	221		BNE BITPERR	
OCDE		222	CHKDVALU	EGU *	
OCDE	C8	223		INY	: Y=1
OCDE	B1 83	224		LDA (VARPN), Y	: GET DELTA LOW
OCDE	90 04	225		BCC NOCOMPL	: DONT COMPLEMENT
OCED	49 FF	226		EDR #FF	
OCEF	69 00	227		ADC #0	: DO TWOS COMP
OCF1	30 F1	228	NOCOMPL	BMI DSIZERR	: DELTA>127 ERR
OCF3	F0 EF	229		BEG DSIZERR	
OCF5	B5 9D	230		STA DELTA	: SAVE WINDOW SIZE
OCF7	A2 02	231		LDX #NNAME	: =2
OCF9	20 E4 0E	232		JSR SETNAME	
OCFC	20 58 0F	233		JSR FINDVAR	: IS NX THERE
OD01	B0 04	234		BCS NTHERE	: YES HE DID
OD01	A2 48	235		PAGE	
OD01	BD 9E	236	NDEFERR	LDX #XINDXDEF	
OD03	BD 9E	237		BNE BITPERR	
OD05		238	NTHERE	EGU *	
OD05	A0 00	239		LDY #0	
OD07	B1 83	240		LDA (VARPN), Y	
OD09	30 35	241		BMI NRNGERR	: INDX<0 ERR
OD0B	B5 9F	242	IXOK	STA INDX+1	: SAVE VALUE
OD0D	C8	243		INY	: Y=1
OD0E	B1 83	244		LDA (VARPN), Y	
OD10	B5 9E	245		STA INDX	
OD12	A5 B3	246		LDA VARPNT	
OD14	B5 A0	247		STA NADRS	: SAVE IT ADDRESS
OD16	A5 B4	248		LDA VARPNT+1	
OD18	B5 A1	249		STA NADRS+1	: FOR UPDATING N
OD1A	A2 04	250		LDX #XNAME	: =4
OD1C	20 E4 0E	251		JSR SETNAME	
OD1F	20 27 0F	252		JSR FINDARY	: IS XX!
OD22	B0 05	253		BCS X THERE	: YES SMART USER
OD24	A2 57	254	ARYERR	LDX #XARYDEF	
OD26	4C A3 OC	255	ERRJMP	JMP BITPERR	
OD29		256	X THERE	EGU *	
OD29	A0 04	257		LDY #4	: POINT AT # OF DIMS
OD2B	B1 83	258		LDA (VARPN), Y	: IS DIMS 1?
OD2D	C9 01	259		CMP #1	: IS IT A VECTOR ?
OD2F	BD F3	260		BNE ARYERR	

0B31	CB	261	XDIM90K	INY	: Y=5
0B32	C9	262		INY	: Y=6
0B33	CB	263		SEC	
0B34	B1 B3	264		LDA (VARPNT), Y	
0B36	E5 9E	265		SBC INDX	
0B38	AA	266		TAX	: SAVE LOW RESULT
0B39	BD	267		DEY	: Y=5
0B3A	B1 B3	268		LDA (VARPNT), Y	
0B3C	E5 9F	269		SBC INDX+1	
0B3E	B0 04	270		BCS NGTXROWS	
0B40	A2 66	271	NRNGERR	LDX #XNRANGE	
0B42	B0 E2	272		BNE ERJUMP	
0B44	30 FA	273	NGTXROWS	BMI NRNGERR	
0B46	B0 04	274		BNE NISOK	
0B48	E0 02	275		CPX #2	
0B4A	90 F4	276		BCC NRNGERR	
0B4C	B1 B3	277	NISOK	LDA (VARPNT), Y	: GET MAX
0B4E	B5 A3	278		STA MAXN+1	
0B50	CB	279		INY	: Y=6
0B51	B1 B3	280		LDA (VARPNT), Y	: GET LO
0B53	B5 A2	281		STA MAXN	
0B55	A5 B3	282		LDA VARPNT	
0B57	18	283		CLC	
0B58	69 07	284		ADC #7	: CALC BASE ADDRESS
0B5A	B5 A6	285		STA XVPTR	: SET X BASE
0B5C	A6 B4	286		LDX VARPNT+1	: GET HI BASE
0B5E	90 01	287		BCC #+3	: NO CARRY FROM ADD
0B60	E8	288		INX	
0B61	B6 A7	289		STX XVPTR+1	: X BASE COMPLETE
0B63	290			PAGE	
0B63	A2 06	291		LDX #YNAME	+ =6
0B65	20 E4 0E	292		JSR SETNAME	
0B68	20 27 0F	293		JSR FINDARY	
0B6B	90 B7	294		BCC ARYERR	
0B6D	A0 04	295	Y THERE	LDY #4	
0B6F	B1 B3	296		LDA (VARPNT), Y	
0B71	C9 01	297		CMP #1	
0B73	D0 AF	298		BNE ARYERR	
0B75	CB	299		INY	
0B76	CB	300		INY	: POINT TO NROWSL
0B77	B1 B3	301		LDA (VARPNT), Y	
0B79	AA	302		TAX	
0B7A	C5 A2	303		CMP MAXN	
0B7C	88	304		DEY	
0B7D	B1 B3	305		LDA (VARPNT), Y	
0B7F	E5 A3	306		SBC MAXN+1	: IS YSIZ>XSIZ?
0B81	B0 1C	307		BCS YSIZOK	: YSIZ>XSIZ
0B83	B6 A2	308		STX MAXN	
0B85	B1 B3	309		LDA (VARPNT), Y	: GET HI BACK
0B87	B5 A3	310		STA MAXN+1	
0B89	38	311		SEC	
0B8A	A5 A2	312		LDA MAXN	
0B8C	E5 9E	313		SBC INDX	
0B8E	AA	314		TAX	
0B8F	A5 A3	315		LDA MAXN+1	
0B91	E5 9F	316		SBC INDX+1	
0B93	B0 02	317		BCS YROWSGTN	
0B95	90 A9	318		BCC NRNGERR	
0B97	30 A7	319	YROWSGTN	BMI NRNGERR	
0B99	B0 04	320		BNE YSIZOK	
0B9B	E0 02	321		CPX #2	: AT LEAST TWO MORE ARRAY EN
0B9D	90 A1	322		BCC NRNGERR	: YSIZON
0B9F	323	YSIZOK		EQU *	
0B9F	A5 B3	324		LDA VARPNT	
0DA1	18	325		CLC	
0DA2	69 07	326		ADC #7	
0DA4	B5 A8	327		STA YVPT	
0DA6	A6 B4	328		LDX VARPNT+1	
0DAB	90 01	329		BCC #+3	
0DAB	E8	330		INX	
0DAB	B6 A9	331		STX YVPT+1	
0DAD	06 9E	332		ASL INDX	
0DAE	26 9F	333		ROL INDX+1	INDX=INDX*2

0DB1	A5 A8	334		LDA YVPT	
0DB3	A6 A9	335		LDX YVPT+1	
0DB5	20 F0 0E	336		JSR ADDINX	
0DB6	B4 A8	337		STY YVPT	
0DB8	B5 A9	338		STA YVPT+1	
0DBC	A5 A6	339		LDA XVPTR	
0DBE	A6 A7	340		LDX XVPTR+1	
0DC0	20 F0 0E	341		JSR ADDINX	
0DC3	B4 A6	342		STY XVPTR	
0DC5	B5 A7	343		STA XVPTR+1	
0DC7	46 9F	344		LSR INDX+1	
0DC9	66 9E	345		ROR INDX	: INDX=INDX/2
0DCB		346		PAGE	
0DCB	A9 00	347		LDA #0	
0DCD	B0 BC 02	348		STA RTNCD	
0DD0	AC 9B 02	349		LDY SAVSL0T	
0DD3	98	350		TYA	
0DD4	38	351		SEC	
0DD5	E9 C1	352		SBC #C1	
0DD7	C9 07	353		CMP #7	
0DD9	B0 08	354		BCS BADMODE	
0DBB	B9 B8 03	355		LDA PAGE, Y	
0DE0	10 06	356		BPL BADMODE	
0DE2	29 7F	357		AND #7F	
0DE4	C9 20	358		CMP WHIRE51	
0DE6	B0 05	359		BCS MODEOK	
0DE6	A2 73	360		LDX #XNOTHGR	
0DE8	AC A3 0C	361		JMP DITPERR	
0DEB	29 03	362		MODEOK	
0DED	AC 16 0C	363		LDY MAXY	
0DF0	AA	364		TAX	: IS IT MIXED MODE?
0DF1	F0 06	365		BEG NOTMIXD	
0DF3	C0 A0	366		CPY #MXYVALU	
0DF5	90 02	367		BCC NOTMIXD	
0DF7	A0 A0	368		LDY #MXYVALU	
0DF9	B4 AA	369		STY TMAXY	
0DFB	A2 02	370		LDX #POINT	
0DFD	BE 9A 02	371		STX SAVSL0T-1	
0E00	20 E1 0E	372		JSR JSRINDRCT	
0E03	AD B0 02	373		LDA TEM	
0E06	29 03	374		AND #3	
0E08	C9 03	375		CMP #3	: IS PEN UP?
0E0A	F0 0A	376		BEG KYBDXIT	: YES, HE HIT KEYBD
0E0C	20 FB 0E	377		JSR WINDOCHK	: IS IT ON SCREEN?
0E0F	B0 06	378		BCS ONSCRN	: YEP
0E11	A9 01	379		LDA #1	: PEN OFF SCREEN EXIT
0E13	BD BC 02	380		STA RTNCD	
0E16	60	381		RTS	
0E17		382		ONSCRN	
0E17	AD 87 02	383		EQU *	
0E1A	AE B5 02	384		LDA TEMYL	
0E1D	AC 86 02	385		LDX TEMXL	
0E20	20 57 F4	386		LDY TEMX	
0E23		387		JSR HPL0T	
0E23	24 A5	388		EQU *	
0E25	30 0B	389		BIT TICFLQ	
0E27		390		BMI NOTICK	
0E27	AD 30 CO	391		EQU *	
0E2A	A9 0F	392		LDA SPKR	
0E2C	20 AB FC	393		JSR WAIT	
0E2F	AD 30 CO	394		LDA SPKR	
0E32		395		NOTICK	
0E32	AD 86 02	396		EQU *	
0E35	AD 00	397		LDY #0	
0E37	91 A6	398		STA (XVPT), Y	
0E39	AD B8 02	399		LDA TEMY	
0E3C	91 A8	400		STA (YVPT), Y	
0E3E	CB	401		INY	
0E3F	AD B5 02	402		LDA TEMXL	
0E42	91 A6	403		STA (XVPT), Y	
0E44	AD 87 02	404		LDA TEMYL	
0E47	91 A8	405		STA (YVPT), Y	

```

OE49: 406 PAGE
OE49: A2 01 407 LDX #1
OE4B: E6 A6 408 XVINC INC XVPTR
OE4D: D0 02 409 BNE **+4
OE4F: E6 A7 410 INC XVPTR+1
OE51: CA 411 DEX
OE52: F0 F7 412 BEQ XVINC
OE54: A2 01 413 LDX #1
OE56: E6 A8 414 YVINC INC YVPTR
OE58: D0 02 415 BNE **+4
OE5A: E6 A9 416 INC YVPTR+1
OE5C: CA 417 DEX
OE5D: F0 F7 418 BEQ YVINC
OE5F: E6 9E 419 INC INDX
OE61: D0 02 420 BNE **+4
OE63: E6 9F 421 INC INDX+1
OE65: 422 * : Y=1 FROM ABOVE!
OE65: A5 9E 423 LDA INDX
OE67: 91 A0 424 STA (NADRS), Y
OE69: 88 425 DEY : Y=0
OE6A: A5 9F 426 LDA INDX+1
OE6C: 91 A0 427 STA (NADRS), Y
OE6E: A5 9E 428 LDA INDX
OE70: C5 A2 429 CMP MAXN
OE72: A5 9F 430 LDA INDX+1
OE74: E5 A3 431 SBC MAXN+1 : IS NCMAX ?
OE76: 90 05 432 BCC MORPTS : YES
OE78: A9 03 433 LDA #3 : ARRAY DURLFD=3
OE7A: 4C 13 0E 434 JMP STRTNCD : GO EXIT
OE7D: AE 9B 02 435 MORPTS LDX SAVSLOT
OE80: BE FB 07 436 STX MSLOT : TO BE SURE!
OE83: 20 F9 CE 437 WAITLP EQU *
OE83: 20 F9 CE 438 JSR MREAD
OE86: AD 80 02 439 LDA TEM
OE89: 29 03 440 AND #3 : IS PEN DOWN?
OE8B: F0 05 441 BEQ PENDOWN : YES
OE8D: A9 02 442 LDA #2 : PEN UP RTN CODE
OE8F: 4C 13 0E 443 JMP STRTNCD
OE92: A2 03 444 PENDOWN LDX #3
OE94: BD B1 02 445 MVLP LDA XFLL, X
OE97: 9D B5 02 446 STA TEMXL, X
OE9A: CA 447 DEX
OE9B: 10 F7 448 SPL MVLP
OE9D: AE 9B 02 449 LDX SAVSLOT
OEAO: 20 FC CE 450 JSR SCALE : SCALE RESULTS!
OEAB: 20 FB 0E 451 JSR WINDOCHK : IS IT ON SCREEN
OEAB: B0 03 452 BCS CKDLTA : YES SO GO ON
OEAB: 4C 11 0E 453 JMP YTODBIG : YES EXIT
OEAB: 454 CKDLTA EQU * : IS NEW POINT IN
OEAB: AD B7 02 455 LDA TEMYL : THE WINDOW ?
OEAE: 3B 456 SEC
OEAF: E5 E2 457 SBC YO : PREVIOUS Y
OE81: B0 04 458 BCS CKYDLTA
OE83: 49 FF 459 EOR #$FF
OE85: 69 01 460 ADC #1
OE87: C5 9D 461 CKYDLTA CMP DELTA : IN WINDOW?
OE89: B0 17 462 BCS PLOTSEQ : NO SO DO IT
OE8B: 3B 463 SEC
OEBC: AD B5 02 464 LDA TEMXL
OE8F: E5 E0 465 SBC XOL : X PREVIOUS
OECD: AA 466 TAX
OE82: AD B6 02 467 LDA TEMX
OE85: E5 E1 468 SBC XOH
OE87: B4 469 TXA
OE8B: B0 04 470 BCS CKXDLTA
OECA: 49 FF 471 EOR #$FF
OECC: 69 01 472 ADC #1
OECE: 473 CKXDLTA EQU *
OECE: C5 9D 474 CMP DELTA
OE80: 90 B1 475 BCC WAITLP : WAIT TILL PEN MOVES
OE82: AD B5 02 476 PLOTSEQ LDA TEMXL
OE85: AE B6 02 477 LDX TEMX
OE8B: AC B7 02 478 LDY TEMYL

```

```

OE8B: 20 3A F5 479 JSR HILIN PLOT LINE SEQ
OE8E: 4C 23 0E 480 JMP MAINLP
OE8E: 481 ****
OE8E: 482 * MY SUBROUTINES
OE8E: 483 ****
OE8E: 484 JSRINDRCT JMP (SAVSLOT-1)
OE8E: 485 ****
OE8E: 486 SETNAME LDA VNAMTAB, X
OE8E: 487 STA VARNAM
OE8E: 488 INX
OE8E: 489 LDA VNAMTAB, X
OE8E: 490 STA VARNAM+1
OE8E: 491 RTS
OE8E: 492 ****
OE8E: 493 ADDINX CLC
OE8E: 494 ADC INDX
OE8E: 495 TAY
OE8E: 496 TXA
OE8E: 497 ADC INDX+1
OE8E: 498 RTS
OE8E: 499 ****
OE8E: 500 WINDOCHK LDA TEMXL : SAVE IT
OE8E: AA 501 TAX
OE8E: CD 11 0C 502 CMP MINXL
OE8E: AD B6 02 503 LDA TEMX
OE8E: AB 504 TAY
OE8E: ED 12 0C 505 SBC MINX : IS X < MIN ?
OE8E: 90 0B 506 BCC WCHKRTS : NO OUTSIDE
OE8E: BA 507 TXA : GET LO BACK
OE8E: CD 13 0C 508 CMP MAXXL
OE8E: 98 509 TYA : GET HI BACK
OE8E: ED 14 0C 510 SBC MAXX : IS XC MAXX
OE8E: 90 02 511 BCC XIN SIDE : YES X IN SIDE
OE8E: 1B 512 OUTSIDE CLC : C=0 SAYS NO
OE8E: 60 513 WCHKRTS RTS
OE8E: AD B8 02 514 XIN SIDE LDA TEMY : ADDED 4/25/79 JOA
OE8E: D0 F9 515 BNE OUTSIDE : ADDED 4/25/79 JOA
OE8E: AD B7 02 516 LDA TEMYL : CHANGED 4/25/79 JOA
OE8E: CD 15 0C 517 CMP MINY : IS Y < MIN Y ?
OE8E: 90 F2 518 BCC WCHKRTS : NO OUTSIDE
OE8E: C5 AA 519 CMP TMAXY : IS Y < MAX Y ?
OE8E: B0 ED 520 BCS OUTSIDE : NO OUTSIDE
OE8E: 3B 521 INSIDE SEC
OE8E: 60 522 RTS
OE8E: 523 PAGE
OE8E: 524 ****
OE8E: A6 6B 525 FINDARY LDX ARYTAB
OE8E: A5 6C 526 LDA ARYTAB+1
OE8E: 86 9B 527 FNDLPA STX LOWTR
OE8E: B5 9C 528 STA LOWTR+1
OE8E: C5 6E 529 CMP STREND+1
OE8E: D0 04 530 BNE FNDFDV
OE8E: E4 6D 531 CPX STREND
OE8E: F0 5C 532 BEQ NOTFND
OE8E: A0 00 533 FNDFDV LDY #0
OE8E: B1 9B 534 LDA (LOWTR), Y
OE8E: C8 535 INY
OE8E: C5 B1 536 CMP VARNAM
OE8E: D0 06 537 BNE NXTARY
OE8E: A5 B2 538 LDA VARNAM+1
OE8E: D1 9B 539 CMP (LOWTR), Y
OE8E: F0 0E 540 BEQ GOTARY
OE8E: C8 541 NXTARY INY
OE8E: B1 9B 542 LDA (LOWTR), Y
OE8E: 1B 543 CLC
OE8E: 65 9B 544 ADC LOWTR
OE8E: AA 545 TAX
OE8E: C8 546 INY
OE8E: B1 9B 547 LDA (LOWTR), Y
OE8E: 65 9C 548 ADC LOWTR+1
OE8E: 90 D7 549 BCC FNDLPA
OE8E: A9 00 550 GOTARY LDA #0
OE8E: F0 2D 551 BEQ ADJVPTR

```

```

OF58: 552 PAGE
OF58: 553 *****
OF58: A5 69 554 FINDVAR LDA VARTAB
OF5A: A6 6A 555 LDX VARTAB+1
OF5C: A0 00 556 LDY #0
OF5E: B6 9C 557 FNDLPX STX LOWTR+1
OF60: B5 9B 558 FINDLP STA LOWTR
OF62: E4 6C 559 CPX ARYTAB+1
OF64: D0 04 560 BNE LOPFN
OF66: C5 6B 561 CMP ARYTAB
OF68: F0 29 562 BEQ NOTFND
OF6A: A5 81 563 LOPFN LDA VARNAM
OF6C: D1 9B 564 CMP (LOWTR),Y
OF6E: D0 08 565 BNE NOTIT
OF70: A5 B2 566 LDA VARNAM+1
OF72: C8 567 INY
OF73: D1 9B 568 CMP (LOWTR),Y
OF75: F0 0C 569 BEQ FOUNDS
OF77: B8 570 DEY
OF78: 18 571 NOTIT CLC
OF79: A5 9B 572 LDA LOWTR
OF7B: B9 07 573 ADC #7
OF7D: 90 E1 574 BCC FNDLPX
OF7F: EB 575 INX
OF80: D0 DC 576 BNE FNDLPX
OF82: 00 577 BRK
OF83: A7 02 578 FOUNDS LDA #2
OF85: 18 579 ADJVPTR CLC
OF86: B5 9B 580 ADC LOWTR
OF88: A4 9C 581 LDY LOWTR+1
OF8A: 90 01 582 BCC ADJDONE
OF8C: C8 583 INY
OF8D: B5 B3 584 ADJDONE STA VARPNT
OF8F: B4 B4 585 STY VARPNT+1
OF91: 38 586 SEC
OF92: 60 587 RTS
OF93: 18 588 NOTFND CLC
OF94: 60 589 RTS
OF95: 590 *****
OF95: 591 * GET USR NAMES FROM*
OF95: 592 * HIS CALL LINE AND *
OF95: 593 * USE THEM INSTEAD *
OF95: 594 * OF THE DEFAULTS *
OF95: 595 * THE USER ENTERS *
OF95: 596 * THE NAMES IN FIXED*
OF95: 597 * POSITIONAL ORDER *
OF95: 598 * AS FOLLOWS *
OF95: 599 * D%, N%, X%, Y% *
OF95: 600 *****
OF95: 601 PAGE
OF95: 602 USRNAMS EQU *
OF95: A2 07 603 LDX #7
OF97: B0 F8 0F 604 DFLTS LDA DEFALT, X
OF9A: 90 09 0C 605 STA VNAMTAB, X
OF9B: CA 606 DEX
OF9E: 10 F7 607 BPL DFLTS
OF9A0: 20 B7 00 608 JSR CHRGOT
OF9A3: D0 01 609 BNE *+3
OF9A5: 60 610 RTS
OF9A6: 20 7D E0 611 JSR ISLETC
OF9A9: 90 1B 612 BCC GETLTR
OF9A8: 20 E7 0F 613 SET1ST JSR STORIT
OF9A8: F0 31 614 BEQ UNAMRTS
OF9B0: 20 F0 0F 615 ISTAIL JSR MYCHGET
OF9B3: 90 1E 616 BCC SET2ND
OF9B5: 20 7D E0 617 JSR ISLETC
OF9B8: B0 19 618 BCS SET2ND
OF9B8: C9 2C 619 CMP #92C
OF9B8: D0 F2 620 BNE ISTAIL
OF9B8: E8 621 NXTX INX
OF9B8: E0 07 622 CPX #7
OF9C1: B0 1E 623 BCS UNAMRTS
OF9C3: 20 F0 0F 624 GETLTR JSR MYCHGET

```

; X, A SET UP
 ; DO CARRY TO HI
 ; X, A SET NOW
 ; INSURANCE!

; GET NEXT CHR
 ; EXIT, END OF STMT
 ; A LETTER?
 ; NO, IGNORE IT
 ; YES, USE

; GET NEXT
 ; DIGIT OK
 ; A LETTER?
 ; YES, USE
 ; NO, A COMMA?
 ; NOT COMMA, IGNORE
 ; A COMMA SAYS NO 2ND
 ; AM I DONE?
 ; YES
 ; FIND A LTR

```

OF66: 20 7D E0 625 JSR ISLETC
OF69: B0 E0 626 BCS SET1ST
OF6B: C9 2C 627 CMP #92C
OF6D: D0 F4 628 BNE GETLTR
OF6F: EB 629 INX
OF70: B8 630 CLV
OF71: 90 EB 631 BVC NXTX
OF73: 20 E7 0F 632 SET2ND JSR STORIT
OF76: F0 09 633 BEQ UNAMRTS
OF78: 20 F0 0F 634 SCANC JSR MYCHGET
OF7B: C9 2C 635 CMP #92C
OF7D: D0 F9 636 BNE SCANC
OF7F: F0 E2 637 BEQ GETLTR
OF81: 20 F0 0F 638 UNAMRTS JSR MYCHGET
OF84: D0 FB 639 BNE UNAMRTS
OF86: 60 640 RTS
OF87: EB 641 STORIT INX
OF8B: 09 B0 642 ORA #980
OF8A: 9D 09 0C 643 STA VNAMTAB, X
OF8D: E0 07 644 CPX #7
OF8F: 60 645 RTS
OFF0: 20 B1 00 646 MYCHGET JSR CHRGOT
OFF3: D0 02 647 BNE *+4
OFF5: 68 648 PLA
OFF6: 68 649 PLA
OFF7: 60 650 RTS
OFF8: C4 60 651 DEFALT DFB $C4, $80
OFFA: CE 60 652 DFB $CE, $80
OFFC: D8 60 653 DFB $D8, $80
OFFE: D9 60 654 DFB $D9, $80

```

*** SUCCESSFUL ASSEMBLY: NO ERRORS

UTILITIES

```

0000: 1 *
0000: 2 *VARIOUS BIT PAD HI-RES ROUTINES
0000: 3 *BY DAVE M. LINCOLN
0000: 4 *COPYRIGHT APPLE COMPUTER CO.
0000: 5 * JUNE 1979
0000: 6 *
0000: 7 DRG $6000
0000: 8 DBJ $2000
0000: 9 HBASL EQU $00
0000: 10 HBASH EQU $01
0000: 11 HMASK EQU $02
0000: 12 ZTEM EQU $03
0000: 13 XOL EQU $2FF
0000: 14 XOH EQU $2FE
0000: 15 YO EQU $2FD
0000: 16 FLGL EQU $2FC
0000: 17 FLGH EQU $2FB
0000: 18 XL EQU FLGL
0000: 19 XH EQU FLGH
0000: 20 *
0000: 21 *WHITE PICK OFF
0000: 22 *
0000: 23 WHITE TXA      SAVE X, Y AND ZPG
0001: 48 24 PHA
0002: 98 25 TYA
0003: 48 26 PHA
0004: A2 03 27 LDX #$3
0006: B5 00 28 SLP LDA $00, X
0008: 48 29 PHA
0009: CA 30 DEX
000A: 10 FA 31 BPL SLP, ZPG SAVED
000C: AD FF 02 32 LDA XOL SELF MODIFY
000F: BD 55 60 33 STA PKE1 BCC INTO BCS
0012: BD 91 60 34 STA PKE2
0015: A9 00 35 LDA #$00 INIT MY REGS
0017: BD FD 02 36 STA YO TO UPPER LEFT
001A: BD FF 02 37 STA XOL W/NO BITS ON
001D: BD FE 02 38 STA XOH
0020: BD FC 02 39 STA FLGL
0023: BD FB 02 40 STA FLGH
0026: 20 C3 60 41 JSR HPOSN FIRST TIME
0029: AE FF 02 42 BLOOP LDX XOL NEXT TIME ONLY
002C: AC FE 02 43 LDY XOH CHANGE Y &
002F: 20 ED 60 44 JSR XPOS HMASK
0032: B1 00 45 LDA (HBASL), Y DO AN
0034: 25 02 46 AND HMASK HSCRN @ X, Y
0036: B5 03 47 STA ZTEM SAVE BIT
0038: F0 02 48 BEG ITZOFF ???
003A: A9 01 49 LDA #$01 NOPE
003C: 1B 50 ITZOFF CLC YUP
003D: BD FC 02 51 ADC FLGL INC FLG CNTR
0040: BD FC 02 52 STA FLGL BY BIT ON
0043: 29 00 53 AND #$00 !CARRY FROM LOW ADD
0045: BD FB 02 54 ADC FLGH !IS ADDED TO FLGH
0048: BD FB 02 55 STA FLGH
004B: AD FC 02 56 LDA FLGL IS THERE MORE
004E: C9 02 57 CMP #$02 THAN 2 BITS ON
0050: AD FB 02 58 LDA FLGH 2 BYTE TEST
0053: E9 00 59 SBC #$00
0055: 90 0A 60 PKE1 BCC LNI30 CS FOR DEL WHITE
0057: A5 03 61 LDA ZTEM BIT?
0059: D0 06 62 BNE LNI30 ON?
005B: 20 02 61 63 JSR CLER NO. OFF
005E: 4C 73 60 64 JMP NXTX NEXT HORIZ POS
0061: AD FC 02 65 LNI30 LDA FLGL IS ANY DN?
0064: 0D FB 02 66 ORA FLGH
0067: F0 04 67 BEQ ZFG NO
0069: A5 03 68 LDA ZTEM YES AND IF
006B: D0 06 69 BNE NXTX Z=1 THEN OK

```

```

606D: BD FC 02 70 ZFG STA FLGL ELSE ZERO
6070: BD FB 02 71 STA FLGH FLAG
6073: EE FF 02 72 NXTX INC XOL INC CNT FIRST
6076: D0 03 73 BNE TESTX THEN TEST
6078: EE FE 02 74 INC XOH X HIGH
607B: AD FF 02 75 TESTX LDA XOL 2 BYTE TEST
607E: C9 18 76 CMP #$18 FOR XPOS=
6080: AD FE 02 77 LDA XOH 279 OR
6083: E9 01 78 SBC #$01 $118
6085: 90 A2 79 BCC BLOOP IF OK ELSE
6087: AD FC 02 80 LDA FLGL END OF X LOOP
608A: C9 02 81 CMP #$02 MAKE SURE TO
608C: AD FB 02 82 LDA FLGH CATCH LAST
608F: E9 00 83 SBC #$00 WHITE OR COLOR
6091: 90 03 84 PKE2 BCC NXTY CS FOR DEL COLOR
6093: 20 02 61 85 JSR CLER CLEAR OUT
6096: A9 00 86 NXTY LDA #$00 RESET X AND
6098: BD FC 02 87 STA FLGL FLAG
609B: BD FB 02 88 STA FLGH
609E: BD FF 02 89 STA XOL
60A1: BD FE 02 90 STA XOH
60A4: EE FD 02 91 INC YO THEN INC YO
60A7: 20 C3 60 92 JSR HPOSN REPOSN
60AA: AD FD 02 93 LDA YO AND TEST
60AD: C9 C0 94 CMP #$CO FOR Y=191
60AF: B0 03 95 BCS RET1 WE'RE DONE
60B1: 4C 29 60 96 JMP BLOOP NO CONTINUE (LONG BRANCH)
60B4: A2 00 97 RET1 LDX #$00 BRING BACK ALL
60B6: 68 98 RLP PLA ZPAGE
60B7: 95 00 99 STA $00, X AND X, Y
60B9: E8 100 INX
60BA: E0 04 101 CPX #$04
60BC: D0 FB 102 BNE RLP
60BE: 68 103 PLA
60BF: A8 104 TAY
60C0: 68 105 PLA
60C1: AA 106 TAX
60C2: 60 107 RTS
60C3: 108 * 109 *HPOSN HRES BIT POSN ROUT
60C3: 110 *READ ABOUT THIS CODE IN UTILITY ROM MANUAL
60C3: 111 * 112 HPOSN LDA YO
60C6: AE FF 02 113 LDX XOL
60C9: AC FE 02 114 LDY XOH
60CC: 48 115 HPOS PHA
60CD: 29 C0 116 AND #$CO
60CF: B5 00 117 STA HBASL
60D1: 4A 118 LSR A
60D2: 4A 119 LSR A
60D3: 05 00 120 DRA HBASL
60D5: B5 00 121 STA HBASL
60D7: 68 122 PLA
60D8: B5 01 123 STA HBASH
60DA: 0A 124 ASL A
60DB: 0A 125 ASL A
60DC: 0A 126 ASL A
60DD: 26 01 127 ROL HBASH
60DF: 0A 128 ASL A
60E0: 26 01 129 ROL HBASH
60E2: 0A 130 ASL A
60E3: 68 00 131 ROR HBASL
60E5: A8 01 132 LDA HBASH
60E7: 29 1F 133 AND #$1F
60E9: 09 40 134 DRA #$40
60EB: B5 01 135 STA HBASH
60ED: BA 136 XPOS TXA THIS ENTRY ONLY
60EE: C0 00 137 CPY #$0 COMPUTES Y & HMASK
60F0: F0 05 138 BEQ HPOSN2
60F2: A0 23 139 LDY #$23
60F4: 69 04 140 ADC #$4
60F6: C8 141 HPOSN1 INY

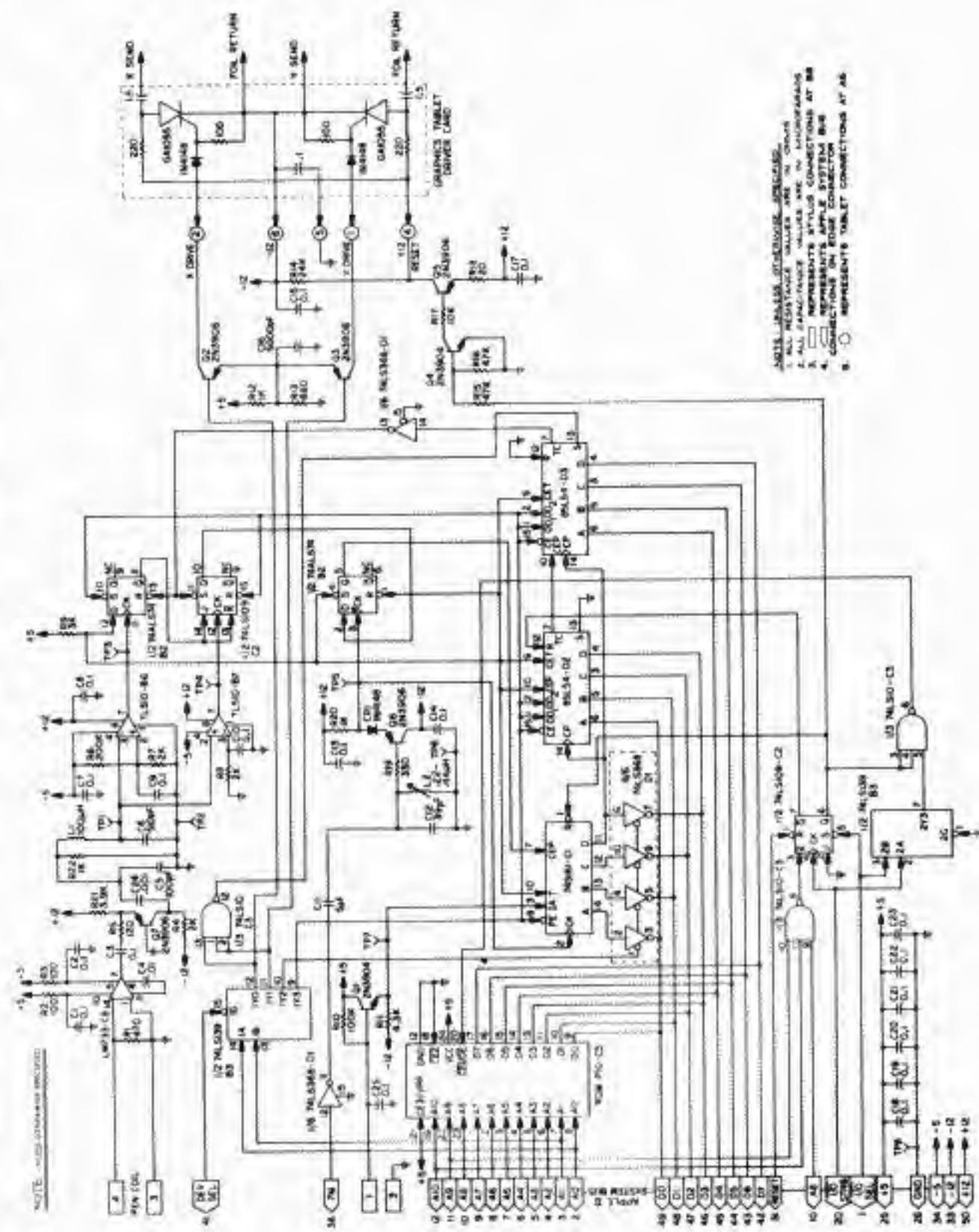
```

60F7	E9 07	142	HP0SN2	SBC #\$7	
60F9	80 FB	143		BCS HP0SN1	
60FB	AA	144		TAX	
60FC	BD 4F 60	145		LDA MSKTBL-249, X	
60FF	85 02	146		STA HMASK	
6101	60	147		RTS	
6102		148 *			
6102		149	*CLER CLEAR BACK FLQ+1 BITS		
6102		150 *			
6102	38	151	CLER	SEC	FIND XO-FLG
6103	AD FF 02	152		LDA XOL	DOUBLE BYTE
6106	ED FC 02	153		SBC FLGL	
6109	BD FC 02	154		STA XL	
610C	AD FE 02	155		LDA XOH	
610F	ED FB 02	156		SBC FLGH	
6112	BD FB 02	157		STA XH	WITH RESULT IN XL, XH
6115	AE FC 02	158	CLOP	LDX XL	SETUP FOR XPOSN
6118	AC FB 02	159		LDY XH	
611B	20 ED 60	160		JSR XPOS	
611E	A5 02	161		LDA HMASK	DELETE THE BIT
6120	49 FF	162		EOR #\$FF	@ Y, HMASK
6122	31 00	163		AND (HBASL), Y	ON LINE HBASL
6124	91 00	164		STA (HBASL), Y	
6126	18	165		CLC	
6127	AD FC 02	166		LDA XL	FAKE OUT TEST SO
612A	69 01	167		ADC #\$01	THAT
612C	CD FF 02	168		CMP XOL	WE ONLY GO TO
612F	AD FB 02	169		LDA XH	XO-1
6132	ED FE 02	170		SBC XOH	
6135	EE FC 02	171		INC XL	
6138	80 03	172		BNE XOH	NOW INC XL, XH
613A	EE FB 02	173		INC XH	
613D	90 D6	174	XOK	BCC CLOP	CONTINUE
613F	A9 00	175		LDA #\$00	DONE CLEAR
6141	BD FC 02	176		STA FLGL	FLAG
6144	BD FB 02	177		STA FLGH	
6147	60	178		RTS	AND RETURN
6148		179 *			
6148		180	*MSKTBL HMASK LOOKUP TABLE		
6148		181 *			
614B	01 02 04				
614B	08	182	MSKTBL	DFB \$01, \$02, \$04, \$08	
614C	10 20 40	183		DFB \$10, \$20, \$40	
614F		184		PAGE	
614F		185 *			
614F		186	*HIRES PICK STUPID PICK ROUTINE		
614F		187 *			
614F		188	*ALTM = \$80 FOR COLOR SET 1		
614F		189	*ALTM = \$00 FOR COLOR SET 2		
614F		190	*COLR = \$AA FOR PUR&BLU		
614F		191	*COLR = \$05 FOR CRN&ORG		
614F		192 *			
614F		193	ALTM	EGU \$2FE	
614F		194	COLR	EGU \$2FF	
614F		195	A1L	EGU \$00	
614F		196	A1H	EGU \$01	
614F		197 *			
614F		198	*HIRES PICK OFF		
614F		199 *			
6150	98	200	HPICK	TYA	SAVE Y, X AND ZPG
6150	48	201		PHA	
6151	A5 00	202		LDA A1L	\$00
6153	48	203		PHA	
6154	A5 01	204		LDA A1H	\$01
6156	48	205		PHA	
6157	A0 00	206		LDY #\$00	SET INDEX
6159	B4 00	207		STY A1L	SET ZPG CTRS
615B	A7 40	208		LDA #\$40	
615D	B5 01	209		STA A1H	TO BEGINNING OF
615F	B1 00	210	LOOP	LDA (A1L), Y	PICK UP BYTE
6161	4D FE 02	211		EOR ALTM	CHANGE SETBIT
6164	30 04	212		BMI QKBYT	IF RIGHT SET

6166	A9 00	213		LDA #00	WRONG SET CLR BYT
6168	F0 0C	214		BEG STOR1	ALWAYS TAKEN
616A	A5 00	215	QKBYT	LDA A1L	ARE ON DDD OR EVEN
616C	4A	216		LSR A	BYTE? THE CARRY KNOWS
616D	AD FF 02	217		LDA COLR	IF EVEN THEN
6170	B0 02	218		BCS STOR	SHIFT ELSE STOR
6172	49 7F	219		EOR #\$7F	SHIFT MASK
6174	31 00	220	STOR	AND (A1L), Y	WIPE OUT EXTRA
6176	91 00	221	STOR1	STA (A1L), Y	AND STORE IT
6178	A5 00	222		LDA A1L	GOTO NEXT BYTE
617A	C9 FF	223		CMP #\$FF	WITH TEST FOR
617C	A5 01	224		LDA A1H	END OF HSCRN
617E	E9 5F	225		SBC #\$5F	(\$5FFF)
6180	E6 00	226		INC A1L	
6182	00 02	227		BNE CHLOP	
6184	E6 01	228		INC A1H	
6186	90 D7	229	CHLOP	BCC LOOP	CC ON NOT END
6188	68	230		PLA	RETURN Y AND ZPG
6189	85 01	231		STA A1H	
618B	68	232		PLA	
618C	85 00	233		STA A1L	
618E	68	234		PLA	
618F	A8	235		TAY	
6190	60	236		RTS	
6191		237		PAGE	

*** SUCCESSFUL ASSEMBLY: NO ERRORS

APPENDIX E SCHEMATIC DIAGRAM





apple computer inc.

10260 Bandley Drive
Cupertino, California 95014
408 996-1010